Technische Universität Ilmenau

Fakultät für Elektrotechnik und Informationstechnik

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

„Zuverlässige Gruppenkommunikation in mobilen Ad-hoc-Netzen auf Basis eines verzögerungstoleranten Kommunikationsdienstes“

vorgelegt von Dipl.-Inf. (FH) Peggy Begerow
geboren am 05. 11. 1972

1. Gutachter: Univ.-Prof. Dr. rer. nat. Jochen Seitz (Betreuer)
2. Gutachter: Univ.-Prof. Dr.-Ing. habil. Kai-Uwe Sattler
3. Gutachter: Prof. Dr.-Ing. Agnieszka Lewandowska

eingereicht am 05. 04. 2016
verteidigt am 03. 11. 2016

urn:nbn:de:gbv:ilm1-2016000562

Danksagung

Diese Arbeit gab mir die Möglichkeit mich selbst weiterzuentwickeln. All dies wäre nicht möglich gewesen, hätten mich nicht zahlreiche Kollegen, Freunde und meine Familie unterstützt.

Die Idee für das Thema ist durch Diskussionen mit Dr. Florian Evers entstanden. Er legte damit den Grundstein für diese Arbeit. Ihm gebührt deshalb besonderen Dank.

Die Verwirklichung dieser Arbeit ermöglichte erst das Gratuierten Kolleg „MOBICOM“ und die damit verbundenen Wissenschaftliche Mitarbeiterstelle im Fachgebiet Kommunikationsnetze der Technischen Universität Ilmenau.

Ein ausdrücklicher Dank gilt Prof. Dr. rer. nat. Jochen Seitz, der mich zuverlässig betreut hat und mir immer beratend zur Seite stand. Weiterhin danke ich Prof. Dr.-Ing. habil. Kai-Uwe Sattler und Prof. Dr.-Ing. Agnieszka Lewandowska, die sich bereit erklärt haben, weitere Gutachten anzufertigen.

Dank an meine Arbeitskollegen, Silvia Krug und Sebastian Schellenberg, die immer sehr viel Geduld bei der Umarbeitung meines englischen Schreibstiels aufbrachten. Silvia Krug half mir, mit ihrer kritischen Sichtweise Probleme zu erkennen und zu lösen.

Ein besonderer Dank, gilt meinem Ehemann Arne Begerow, der mir immer Rückenhalt gegeben hat, mich in allem unterstützt und mich während der ganzen Promotionsphase liebevoll umsorgt hat. Auch möchte ich meinen Eltern danken, die viel Zeit mit meinen Kindern verbracht haben. Da diese oft auf mich verzichten mussten, gebührt auch ihnen besonderer Dank.

Und nicht zuletzt danke ich Ines Richter, die mich bei Grammatik und Rechtschreibung in Bezug auf diese Arbeit tatkräftig unterstützt hat.

Danke an alle Mitarbeiter des Fachgebiets Kommunikationsnetze, die immer in der Lage waren mich mit einem Witz neu zu motivieren. Alle fachlichen Diskussionen, bei denen auch manche Mittagspause herhalten musste, haben zum Erfolg dieser Arbeit beigetragen.

Nochmals herzlichen Dank an Alle, die mich unterstützt haben!

Kurzfassung

Ein zuverlässiges Netz für die Kommunikation ist die Basis für eine erfolgreiche Organisation und Koordination von Rettungskräften in Katastrophenfällen. Die heutige Kommunikationstechnik der Rettungskräfte basiert auf dem digitalen Funksystem Terrestrial Trunked Radio (TETRA). TETRA bietet keine ausreichende Datenrate für Multimediadaten und ist bei zerstörter Infrastruktur nur eingeschränkt nutzbar. Deshalb ist es notwendig die Kommunikation in Katastrophenfällen auf anderen Netztypen aufzubauen und Protokolle weiterzuentwickeln.

Die vorliegende Arbeit befasst sich mit der zuverlässigen Gruppenkommunikation in Katastrophenfällen. Durch die oft fehlende Infrastruktur in solchen Szenarien, werden Mobile Ad-hoc Networks (MANETs) verwendet, um eine Kommunikation kurzfristig wieder herzustellen. MANETs bilden sich selbständig und sind in ihrer Reichweite eingeschränkt. Das kann dazu führen, dass mehrere zu einer Kommunikationsgruppe gehörende Kommunikationspartner nicht direkt miteinander verbunden sind. Um trotzdem eine Kommunikation zu ermöglichen, wurde unter Nutzung eines verzögerungstoleranten Kommunikationsdienstes (Delay Tolerant Networking (DTN)) ein Gruppenkommunikationsprotokoll entwickelt. Dieses Protokoll (Reliable Multicast over Delay Tolerant Mobile Ad Hoc Networks (RMDA)) übermittelt Gruppennachrichten mit einer hohen wählbaren Zuverlässigkeit an die gewünschten Gruppenmitglieder unter Optimierung des Speicherplatzbedarfs der DTN-Knoten.

Abstract

A reliable network for communication is the basis for a successful organization and coordination of rescue services in case of disasters. Today's communication technology of the emergency services is based on the digital radio system Terrestrial Trunked Radio (TETRA). TETRA provides no sufficient data rates for multimedia data. In case of destroyed infrastructure it is available only to a limited extent. Therefore in case of disasters, it is necessary to provide communication services based on other network types and further development of protocols.

This thesis is concerned with reliable group communication in disaster scenarios. By the frequent lack of infrastructure in such scenarios, Mobile Ad-hoc Networks (MANETs) are used to restore quickly. MANETs build themselves autonomously and are locally limited. As a result, group members belonging to one multicast group could be not directly connected. Therefore, to enable a communication, a group communication protocol was developed using a delay-tolerant communication service (Delay Tolerant Networking (DTN)).

This protocol (Reliable Multicast over Delay Tolerant Mobile Ad Hoc Networks (RMDA)) sends group messages with selectable high degree of reliability to the desired group members, while optimizing the buffer required on the DTN nodes.

Inhaltsverzeichnis

1 Einleitung

2011 wurde eine nukleare Katastrophe durch ein Erdbeben und den darauf folgenden Tsunami in Japan ausgelöst. Weite Teile der Küste, mit deren Infrastruktur, wurden zerstört. Trotz schnell eingeleiteten Rettungsmaßnahmen verloren viele Menschen ihr Leben. Durch eine bessere Kommunikation und damit Organisation der Rettungsteams hätten mehr Leben gerettet werden können. Deshalb ist die Forschung auf dem Gebiet der *Zuverlässigen Gruppenkommunikation in mobilen Ad-hoc-Netzen auf Basis eines verzögerungstoleranten Kommunikationsdienstes* in der heutigen Zeit so bedeutend.

1.1 Motivation

In den letzten Jahren traten immer häufiger Naturkatastrophen auf, bei denen tausende Menschen ihr Leben verloren. Beispiele hierfür sind 2015 ein Erdbeben in Nepal sowie im Jahre 2011 in Japan. Menschen in solchen Gebieten benötigen schnelle und effiziente Rettungs- und Hilfsmaßnahmen. Doch die Infrastruktur in den betroffenen Gebieten wurde weitgehend zerstört. Viele Helfer aus verschiedenen Ländern boten Hilfskräfte und Hilfsgüter an. Um dies effizient zu koordinieren ist eine funktionierende Kommunikation notwendig. Das Problem liegt in fehlenden länderübergreifenden Standards für die Kommunikation.

Als erster Schritt muss die Kommunikation wieder hergestellt werden, damit das Leben der Rettungskräfte, z. B. durch radioaktive Strahlung, Flutwellen oder Lawinen, nicht unnötig gefährdet wird. Da keine oder nur teilweise Infrastruktur vorhanden ist, bieten sich Mobile Ad-hoc Networks (MANETs) für den schnellen Aufbau der Kommunikation zwischen Rettungskräften an. Mit Hilfe von mobilen Geräten wie Smartphones Laptops werden eigenständige Netze aufgebaut.

Im zweiten Schritt werden Rettungskräfte verschiedenen Gruppen zugeordnet, wie z. B. Feuerwehr, Polizisten, Sanitäter usw. Mitglieder innerhalb einer Gruppen sind an den gleichen Informationen interessiert. Denkbar ist, dass an die Gruppe „Polizisten“ eine Nachricht gesendet wird, die beinhaltet, dass ein bestimmtes Gebiet abzusperren ist. Diese Nachricht sollte (möglichst) alle Polizisten zuverlässig und (möglichst) schnell erreichen. Deshalb bietet sich insbesondere in diesen Fällen die Gruppenkommunikation an.

Bei großflächigen Katastrophen ist durch die Bewegung der Rettungskräfte und die begrenzte Reichweite mobiler Geräte im Ad-hoc-Modus eine komplette Abdeckung der Kommunikation oft nicht gewährleistet. Demzufolge entstehen oft Unterbrechungen zu verschieden Netzabschnitten.

Aktuell verwenden Behörden und Organisationen mit Sicherheitsaufgaben (BOS) in Deutschland das digitale Funksystem Terrestrial Trunked Radio (TETRA) oder sie nutzen noch immer analoge Funktechniken [Hußmann u. a., 2000]. Der TETRA-Standard ist infrastrukturbasiert, wobei auf einen Ad-hoc-Modus umgestellt werden kann, welcher keine Unterbrechungen toleriert. Um TETRA nutzen zu können, müssen alle Nutzer ein TETRA-fähiges Gerät besitzen. Die Anschaffung solcher Geräte ist kostenintensiv, weshalb viele BOS, also Feuerwehreinheiten und Polizeistationen, noch nicht umgerüstet sind. Des Weiteren erfolgt die Kommunikation von der Einsatzstelle zur Leitstelle weitestgehend über herkömmliche Mobilfunknetze. [Christiansen, 2014] Die Bandbreite des TETRA-Systems ist nur für die Übertragung von Sprachnachrichten sowie Kurznachrichten ausgelegt, und somit nicht für die Übertragung von Multimediadaten geeignet [Wolff, 2013].

Ein verzögerungstoleranter Dienst (Delay Tolerant Networking (DTN)) bietet eine Möglichkeit Nachrichten zu speichern und diese bei Kontakt zu anderen Netzteilen, weiterzuleiten. Auf diesem Weg können Unterbrechungen überbrückt werden. Durch Unterbrechungen verursachte Wissenslücken über die aktuellen Anzahl der Gruppenmitglieder oder auch die fehlende Kenntnis über die Netzstruktur, fordert die Weiterentwicklung von Protokollen, die gezielt Algorithmen bereitstellen, um unter diesen Bedingungen eine Kommunikation zu gewährleisten. Eine der größten Herausforderung besteht darin, dass alle Gruppenmitglieder eine versendete Gruppennachricht erhalten. Dabei dürfen die eingeschränkten Ressourcen, verursacht durch die mobilen Geräte, sowie die Identifikation von Gruppenmitgliedern nicht vernachlässigt werden.

Um eine zuverlässige Gruppenkommunikation, insbesondere bei Katastrophen zu gewährleisten, wird in der vorliegenen Arbeit ein neues Protokoll definiert, über welches Gruppennachrichten mit einer hohen, wählbaren Zuverlässigkeit an alle Gruppenmitglieder unter Optimierung des Speicherplatzbedarfs und unter Nutzung eines verzögerungstoleranten Kommunikationsdienstes übermittelt werden. Eine verbesserte Kommunikation ist die Grundlage für das schnelle Auffinden und Bergen von Opfern. Es hilft Menschenleben zu retten.

1.2 Zielstellung und wissenschaftlicher Beitrag

Ziel dieser Arbeit ist es, ein Protokoll zu entwickeln, welches eine zuverlässige Gruppenkommunikation innerhalb eines Katastrophenszenarios ermöglicht. In der Veröffentlichung von A. Onwuka [Onwuka, 2011] wurde festgestellt, dass MANETs besonders gut geeignet sind, um eine Kommunikation in Katastrophengebieten neu aufzubauen. Deshalb soll das Gruppenkommunikationsprotokoll auf MANETs aufsetzen. Problematisch ist, dass durch die Mobilität der Knoten eine Partitionierung des Netzes entstehen kann, wodurch die Kommunikation unterbrochen wird. Zu diesen MANET-Segmenten ist eine Kommunikation mit herkömmlichen Kommunikationsprotokollen nur schwer oder gar nicht möglich. Zur Verbindung dieser MANET-Segmente, soll ein verzögerungstoleranter Dienst, auch bekannt unter DTN, integriert werden. Hierbei werden die Mobilität von Knoten oder auch speziell eingesetzte Knoten zum Datentransport (Fähren) genutzt, um diese Netzsegmente zu verbinden und so gespeicherte Nachrichten auszutauschen. Gruppenkommunikation ist eine häufige Kommunikationsform in Katastrophenszenarien. Sie dient zur Koordination von Helfereinheiten und sorgt für die Verteilung lebenswichtiger Informationen.

Deshalb soll das Kommunikationsprotokoll den Fokus auf die Zustellung von Gruppennachrichten legen. Besonders wichtig ist darüber hinaus, dass Gruppennachrichten möglichst allen Gruppenmitgliedern zugestellt werden, auch wenn diese gerade nicht direkt zu erreichen sind und sich in einem anderem Netzsegment befinden. Das Ziel des Protokolls ist, die zuverlässige Gruppenkommunikation über ein verzögerungstolerantes mobiles Ad hoc Netz zu realisieren. Bestehende Ansätze werden im Abschnitt 3.3.1 aufgezeigt und diskutiert.

Unter Beachtung des obengenannten Ziels wurde folgendes Gruppenkommunikationsprotokoll Reliable Multicast over Delay Tolerant Mobile Ad Hoc Networks (RMDA) erarbeitet:

- RMDA baut auf dem DTN-Konzept auf, um Unterbrechungstoleranz zu gewährleisten.
- Es besteht aus zwei Modulen, dem Gruppenverwaltungsmodul (Unterkapitel: 4.3) und dem Übertragungsmodul (Unterkapitel: 4.4), welche dezentral in jedem RMDA-fähigen Knoten vorhanden sind und somit den Nachteilen einer zentralen Verwaltung entgegenwirken.
- Gruppen werden über eine End-Point-Identifikation (EID) adressiert. Ein Gruppenbeitritt bzw. ein Gruppenaustritt ist jederzeit möglich.
- Das Protokoll nutzt eine intelligente neuartige Speicherverwaltungsstrategie (Abschnitt: 4.4.3) mit integriertem RMDA-Algorithmus (Abschnitt: 4.4.5), um hohe Auslieferungsraten zu erreichen, und erhöht daher die zuverlässige Übertragung von Gruppennachrichten.
- RMDA erlaubt den Anwendern das Protokoll auf das jeweilige Nachrichtenaufkommen anzupassen, indem es drei Level zur Auswahl stellt. Je nach Level werden das Verhältnis des Nachrichtenspeichers von Gruppennachrichtennachrichten zu Unicastnachrichten im Knoten verändert.
- Die Empfängeridentifikation (Abschnitt: 4.4.2) im Übertragungsmodul bietet eine zusätzliche Empfängerauswahl, abhängig vom Zeitpunkt des Gruppenbeitritts oder Gruppenaustritts.

RMDA nutzt eine neuartige intelligente Speicherverwaltungsstrategie mit einem Schätzungsalgorithmus, dem RMDA-Algorithmus, der trotz unterbrochener Netzsegmente eine verbesserte, und somit zuverlässigere, Zustellung von Gruppennachrichten ermöglicht.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit wird in Kapitel untergliedert. Wichtige Begriffe werden definiert. Nach der Einleitung werden Grundlagen (Kapitel 2) zu den benötigten Netztypen sowie zur Gruppenkommunikation vermittelt. Anschließend erfolgen Anforderungen und Bewertungskriterien von DTNs (Kapitel 3). Dort werden auch bisherige Arbeiten vorgestellt. Der Haupteil, das Konzept von RMDA, wird im darauffolgenden Kapitel 4 erörtert. Es wird dabei detailliert auf die einzelnen Bestandteile von RMDA eingegangen. Im Kapitel 5 werden verschieden Simulationen vorgestellt und diskutiert. Danach wird im Kapitel 6 eine kurze Zusammenfassung dieser Arbeit sowie ein Ausblick auf die möglichen zukünftigen Weiterentwicklungen gegeben.

2 Grundlagen

In diesem Kapitel werden Begriffe definiert, die zum allgemeinen Verständnis dieser Arbeit benötigt werden. Zu Beginn werden Grundlagen zu Netzen (Unterkapitel 2.1) beschrieben, die für diese Arbeit relevant sind. Dabei werden gebräuchliche Definitionen vorgestellt. Sie bilden die Basis für die weiteren Kapitel. Unterkapitel 2.2 erklärt den Vorteil einer Kombination von MANETs und DTNs. Danach werden im Unterkapitel 2.3 Kommunikationsformen vorgestellt. Auf die Gruppenkommunikation wird im Detail eingegangen.

2.1 Netztypen

Um die Kommunikation zwischen verschieden Teilnehmern zu ermöglichen sind Netze (Definition 1) notwendig. Dabei haben sich die Netze von anfänglich einfachen Telefonnetzen bis hin zu modernen Mobilfunknetzen entwickelt.

Definition 1 (Netz) *Ein Netz verbindet verschiedene Knoten (mindestens zwei) physikalisch und logisch so miteinander, dass diese kommunizieren können. Diese Verbindung erfolgt über Kabel und/oder über Funk. Es bietet die Möglichkeit Ressourcen, wie Drucker, zu teilen. [Machajewski, 2015]*

Netze kann man hauptsächlich in zwei Netztypen einteilen: in Infrastrukturnetze (Abschnitt 2.1.1) und Ad-hoc Netze (Abschnitt 2.1.2). Wichtige Eigenschaften, Vor- und Nachteile sowie eine kurze Erklärung zu diesen Netztypen werden im Folgenden vorgestellt.

2.1.1 Infrastrukturnetze

Infrastrukturnetze (Definition 2) sind weltweit der weit verbreitetste Netztyp und bilden die Grundlage unserer heutigen Kommunikation. Herkömmliche Telefonnetze als auch alle Mobilfunknetze basieren auf einer Infrastruktur. Die Kommunikation zwischen Kommunikationspartnern erfolgt immer über einen Zugangspunkt. Ein weiteres klassisches Beispiel eines Infrastrukturnetzes ist das Internet, welches gleichzeitig das bekannteste Wide Area Network (WAN) (Definition 4) ist. Unternehmensinterne Netze gehören zu der Kategorie der Local Area Networks (LANs).

Bei Infrastrukturnetzen entstehen hohe Wartungskosten. Eine Planung beim Aufbau neuer Infrastrukturnetze ist im Vorfeld immer notwendig.

Definition 2 (Infrastrukturnetz) *Infrastrukturnetze sind stationäre Netze mit Routern und Servern oder Mobilfunknetze mit Basisstationen und hierarchischer Struktur. Diese Netze haben einen zentralen Dienstanbieter. [Irmscher, 2011]*

Definition 3 (Local Area Network) *Ein lokales Netz (Local Area Network, LAN) ist ein „Datenkommunikationssystem (Netz), das die Übertragung von Daten zwischen mehreren unabhängigen Datenstationen (v. a. Rechnern) mit hoher Übertragungsgeschwindigkeit und mit niedriger Fehlerrate in einem begrenzten geografischen Gebiet ermöglicht. Es befindet sich i. d. R. im Besitz und Gebrauch einer einzelnen Organisation.“ [Lackes und Siepermann, 2015a]*

Definition 4 (Wide Area Network) *Ein Wide Area Network (WAN), auch Weitverkehrsnetz genannt, ist ein Netz, welches über einen geografisch größeren Raum verteilte Datenstationen (v. a. Rechner) verbindet. [Lackes und Siepermann, 2015b] Ein WAN verbindet verschiedene LANs miteinander. Dabei spielt es keine Rolle, ob die LANs kabelbasierend oder funkbasierend sind.*

2.1.2 Ad-hoc Netze

Ad-hoc Netze (Definition 7) sind heutzutage wenig verbreitet. Ad-hoc Netze besitzen keine feste Infrastruktur. Bei diesem Netztyp ist keine vorherige Netzwerkplanung und keine zentrale Verwaltung nötig. Ein Beispiel für solch ein Netz ist eine Funkverbindung über ein Wireless Local Area Network (WLAN) [Board, 2012] (Definition 5) oder Bluetooth [Board, 2005] (Definition 6) zwischen zwei Smartphones mit dem Zweck des Datenaustauschs.

Definition 5 (WLAN) *„Wireless Local Area Networks (WLANs) sind drahtlose LANs, die ihre Daten mit Funk übertragen. Wenn man von WLANs spricht, meint man die von der Arbeitsgruppe Institute of Electrical and Electronics Engineers (IEEE) 802.11 standardisierten...“ [Lipinski u. a., 2015j] Netze.*

Definition 6 (Bluetooth) *„Bluetooth ist der Standard für die Funkkommunikation mit geringer Reichweite und gehört zu den Technologien von Short Range Wireless (SRW). Die Reichweite liegt bei etwa 10 Metern und ist bedingt durch die festgelegte Sendeleistung von 0 dBm und die hohe Freiraumdämpfung bei der Übertragungsfrequenz von 2,4 GHz. Durch Einsatz von Verstärkern kann die Entfernung auf 100 Meter erhöht werden.“ [Lipinski u. a., 2015b]*

Definition 7 (Ad-hoc Netz) *„Ad-hoc Netze (AHN) sind in sich geschlossene Netzwerke, die sich selbst organisieren und keine Hierarchie haben. Ad-hoc-Netzwerke bauen sich nur für die Dauer der Kommunikation auf, sie besitzen keine festgelegte Kommunikationsstruktur und konfigurieren und organisieren sich selbst. Es sind sehr leistungsfähige Self Organized Networks (SONs) (Definition 8), die sich durch eine gute Lastverteilung auszeichnen und kein zentrales Management besitzen. In solchen Netzkonstellationen übernehmen die Endgeräte das Routing und speichern die Routingtabellen.“ [Lipinski u. a., 2015a]*

Definition 8 (Self Organized Network) *„Sich selbst organisierende Netzwerke, SONs gibt es bei den Ad-hoc-Netzen, den Internetprotokoll (IP)-basierten Peer-to-Peer-Netzen, bei WLANs, Mesh-Netzen und bei Mobilfunknetzen der höheren Generationen. Diese SON-Netze entscheiden nach eigenen Algorithmen über die Kooperation der Knoten mit benachbarten Knoten. Sie entscheiden selbst auf Grund von Erfahrungen, mit wem sie kooperieren und mit wem nicht. Die Erfahrungen und Empfehlungen können wirtschaftlicher, sozialer oder computertechnischer Art sein.“ [Lipinski u. a., 2015i]*

2.1.2.1 MANETs

MANETs (Definition 9) gehören zu den Ad-hoc Netzen. Im Gegensatz zu Internet oder Mobilfunk, die auf Infrastruktur basieren, sind MANETs infrastrukturlos und mit dynamischer Topologie für Einsätze bei Rettungsaktionen in Katastrophengebieten sehr gut geeignet. [Onwuka, 2011] Nachteilig ist die geringere und variable Bandbreite gegenüber kabelbasierten Netzen. Es ist auch davon auszugehen, dass durch die mobilen Geräte Batteriekapazität und Speicherkapazität nur eingeschränkt vorhanden sind.

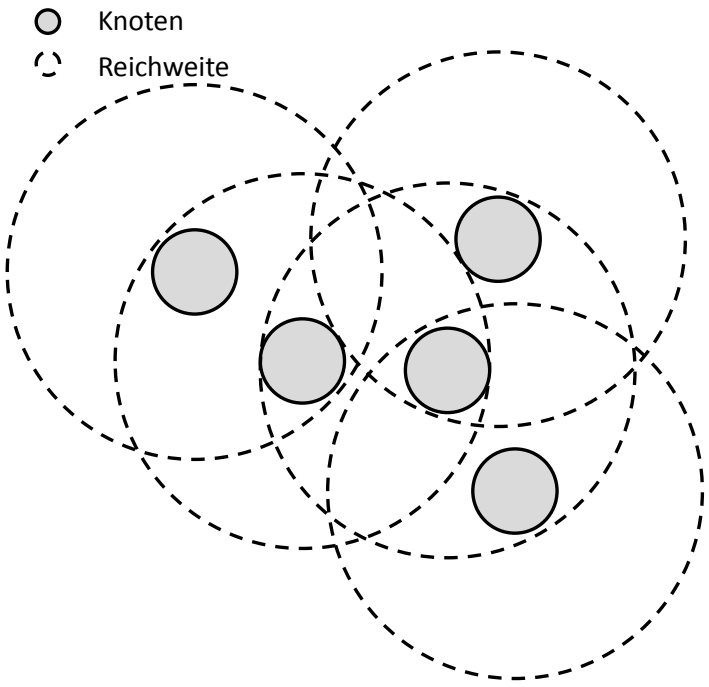

Abbildung 2.1: MANET

Definition 9 (Mobiles Ad-hoc Netz) *Ein Mobile Ad-hoc Network (MANET) ist ein selbst konfigurierendes infrastrukturloses Netz von mobilen Knoten, welche über Funk miteinander verbunden sind. Jeder Knoten kann sich frei in jede Richtung bewegen, was zu einem ständigen Wechsel der Verbindung untereinander führt. Jeder Knoten leitet eigene und fremde Daten weiter und fungiert deshalb als ein Router. [Wikipedia, 2015a]*

Traditionelle MANETs arbeiten unter der Voraussetzung, dass ein Pfad vom Sender und Empfänger immer aufgebaut werden kann, wobei davon ausgegangen wird, dass die Paketverlustrate als auch die Ende-zu-Ende-Verzögerung relativ gering sind. Ein einfaches MANET ist in Abbildung 2.1 dargestellt.

MANETs arbeiten auf verschieden Funkstandards wie WLAN (IEEE 802.11 [Board, 2012]) oder Bluetooth (IEEE 802.15.1 [Board, 2005]). Die Reichweite der mobilen Knoten ist je nach Funkstandard begrenzt.

2.1.2.2 DTNs

DTNs oder auch verzögerungstolerante Netze (Definition 10) sind aus dem von der National Aeronautics and Space Administration (NASA) entwickelten Interplanetaren Internet (IPN) entstanden. Der Fokus beim DTN liegt auf der Verbindung spärlich verbundener heterogener Netze (Definition 11) im Gegensatz zum Interplanetares Internet (IPN), wo der Schwerpunkt bei der Weltraumkommunikation liegt. DTNs sind fähig Verbindungsunterbrechungen zu überbrücken. Durch neue Anwendungsgebiete von DTNs, wie z. B. in Katastrophenfällen, intensiviert die Delay-Tolerant Networking Research Group (DTNRG) dessen Weiterentwicklung seit 2002.

Definition 10 (Delay Tolerant Network) *Ein DTN ist ein Netz, welches Unterbrechungen überbrückt als auch Daten mit hoher Laufzeit über große Entfernungen überträgt. In einem DTN existiert normalerweise keine durchgehende Verbindung vom Sender zum Empfänger. Es arbeitet nach dem Prinzip store, carry und forward (speichern, mit sich tragen, weiterleiten). [Fall, 2003]*

Definition 11 (Heterogenes Netz) *Ein heterogenes Netz besteht aus einen oder mehreren Schichten (Definition 12) aus ungleichen Komponenten. Es kann beispielsweise aus unterschiedlichen Übertragungsmedien bestehen oder auf unterschiedlichen Netztopologie beruhen. [Lipinski u. a., 2015d]*

Definition 12 (Schicht) *„In der Kommunikation werden die Funktionalitäten in einzelnen Schichten eines Schichtenmodells festgeschrieben. Schichtenmodelle gibt es für alle Netzwerkarchitekturen und Netzwerkbetriebssysteme ... Diese Modelle bestehen aus mehreren Schichten, auf denen Elemente mit vergleichbaren Funktionen residieren. Jede Schicht beschreibt die Funktionen der Elemente.“ [Lipinski u. a., 2015h]*

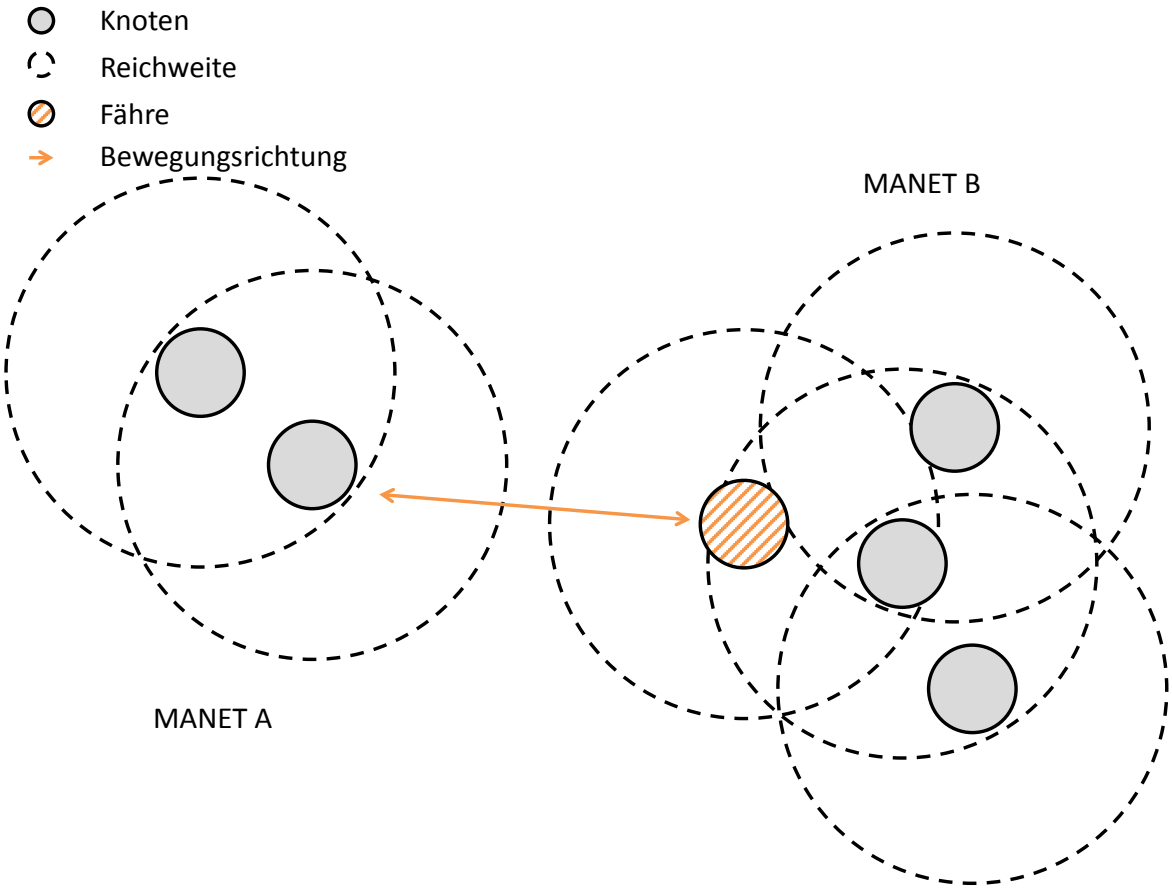

Abbildung 2.2: DTN

Definition 13 (Fähre) *Eine Fähre ist ein spezieller mobiler Knoten, der Kommunikationsdienste für andere Knoten im Netzwerk bereitstellt. Diese Fähre bewegt sich im Einsatzraum und übernimmt den (physischen) Transport von Daten zwischen den Knoten. [Zhao u. a., 2004]*

Wie oben schon erwähnt, bezieht sich die Forschung von DTNs heutzutage hauptsächlich auf heterogene Netze. Ein Beispiel eines DTNs ist in Abbildung 2.2 dargestellt. In diesem Beispiel werden zwei MANETs durch eine Fähre (Definition 13) verbunden. Die Fähre speichert die Nachrichten vom MANET A und transportiert diese physikalisch zum MANET B. Kommt die Fähre mit einem Knoten aus dem MANET B in Kontakt (Definition 14), überträgt die Fähre diese Nachrichten für dieses MANET. MANET B gibt wiederum Nachrichten an die Fähre usw.

In Abbildung 2.3 wird der DTN-Dienst im Internet-Referenzmodell (Definition 15) dargestellt. DTNs können auf zwei Schichten im Internet-Referenzmodell arbeiten: auf der Anwendungsschicht oder in der Netzzugangsschicht.

Definition 14 (Kontakt) *Kontakt zwischen Knoten besteht dann, wenn eine Funkverbindung zwischen diesen Knoten besteht und es möglich ist, Daten auszutauschen.*

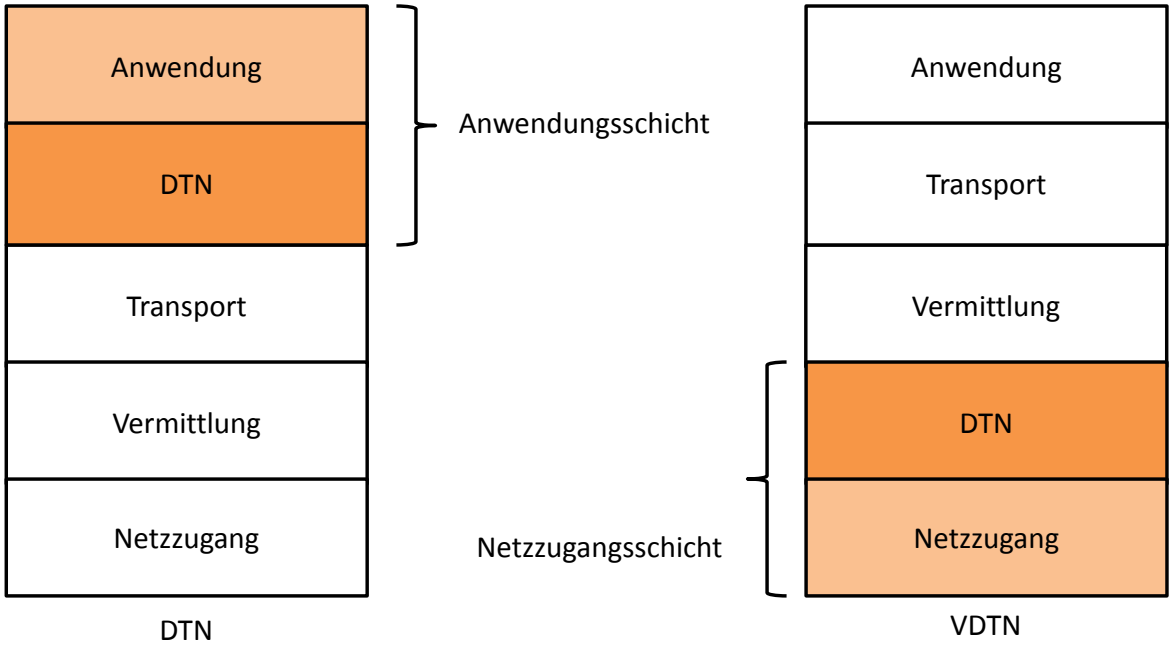

Abbildung 2.3: DTN in verschiedenen Schichten

Definition 15 (Referenzmodell) *Referenzmodelle, wie das Internet-Referenzmodell, dienen als Werkzeug um eine offene Kommunikation zwischen unterschiedlichen Netzwerkgeräten verschiedener Hersteller zu ermöglichen. Die Schichten in einem Referenzmodell sind so festgelegt, dass diese aufeinander aufbauen und über festgelegte Schnittstellen Dienste (Definition 16) austauschen.*

Definition 16 (Dienst) *„Ein Dienst ist in der Open Systems Interconnection (OSI)-Terminologie eine Funktionssammlung einer Schicht, die diese einer übergeordneten Schicht am sogenannten Dienstzugangspunkt anbietet. Ein Dienst wird immer der direkt übergeordneten Schicht angeboten. Die Dienste der einzelnen Schichten werden von den unterschiedlichen Aufgaben dieser Schichten geprägt. So gibt es auf der Netzwerkebene die Netzwerkdienste, auf der Transportschicht die Transportdienste und auf der Anwendungsschicht die Anwendungsdienste.“ [Lipinski u. a., 2015c]*

Definition 17 (Bundle) *Ein Bundle ist die Nachricht, die in einem DTN ausgetauscht wird. Das Format wurde im Requests for Comments (RFC) 5050 [Scott und Burleigh, 2007] festgelegt.*

Vorreiter war das DTN in der Anwendungsebene. Das RFC 4838 [Cerf u. a., 2007] beschreibt die Architektur eines DTNs auf dieser Ebene. Dabei teilt sich die DTN-Schicht in die Teilschicht Bundleebene und in die Teilschicht Anpassungsebene auf. Die Bundleebene ist für die Umwandlung der Nachrichten aus den höheren Schichten in Bundles und deren Adressierung verantwortlich. Die Anpassungsebene dient dabei lediglich zur Anpassung der Bundles an die unterliegenden Schichten. Das Bundle-Protokoll (RFC 5050 [Scott und Burleigh, 2007]) ist das heutige Standardprotokoll und beschreibt die Zusammenfassung von Datenpaketen zu einen Bundle (Datenbündel), welches gleichzeitig alle Daten bzw. Metadaten enthält. Diese Bundles werden in der DTN-typischen Transportweise (Speichern und Weiterleiten) über verschiedene Netzknoten und Netztechnologien schrittweise, auch Hop-bei-Hop genannt, weitergeleitet. Laut [Fall, 2003] erfolgt die Adressierung mittels EID nach dem Uniform Resource Identifier (URI)-Format ([Berners-Lee u. a., 2005]) und wird via spätes Binden (Late Binding) zu einer IP-Adresse gewandelt. Allerdings existiert keine detaillierte Beschreibung, wie das späte Binden erfolgen soll.

Das DTN in der Netzzugangsschicht wird hauptsächlich in Vehicular Delay Tolerant Networks (VDTNs) verwendet und wurde von Soares et al. [Soares u. a., 2009] vorgestellt. DTN/VDTN in dieser Schicht unterstützt Funktionalitäten der Netzzugangsschicht. Ein Beispiel hierfür ist in der Veröffentlichung [Begerow u. a., 2014b] erläutert. Dort wird mittels Informationen (Signalstärke und Bandbreite) aus der Netzzugangsschicht eine Handoverfunktionalität (Definition 19) zum Umschalten in den DTN-Modus vorgestellt.

Definition 18 (Vehicluar Delay Tolerant Network) *Ein Vehicluar Delay Tolerant Network ist eine Erweiterung des DTN. In diesem Netz werden Fahrzeuge (z. B. Autos, Busse und Boote) so genutzt, um einen Nachrichtenübermittlungsdienst anzubieten, der durch die Bewegung der Fahrzeuge im Netz, Nachrichten von Knoten sammelt. [Soares u. a., 2009]*

Definition 19 (Handover) *Ein Handover ist ein Übergabeverfahren einer Kommunikationsverbindung zwischen funkbasierten Netztechniken. Dabei wird zwischen horizontalem Handover (Übergabeverfahren innerhalb einer Netztechnik) und vertikalem Handover (Übergabeverfahren zwischen verschiedenen Netztechniken) unterschieden.*

Diese verschiedenen Netztypen können auch zu einem heterogenen Netz (siehe Definition 11) verbunden werden. Das nachfolgende Kapitel gibt einen kurzen Überblick über Anwendungsfälle und beschreibt, an welchen Stellen Probleme in heterogenen Netzen auftreten können.

2.2 Heterogene Netze

Der Vorteil der Kombination verschiedener Netztypen ist schon länger bekannt. Im Projekt Security System for Public Institutions in Disastrous Emergeny Scenarios (SPIDER) [Wietfeld, 2012] wurde deshalb ein Lösungsvorschlag zur Kombination von verschiedenen Funksystemen zur optimierten Kommunikation von Einsatzkräften in Katastrophenszenarien vorgeschlagen. Dort werden öffentliche Mobilfunknetze, TETRA, satellitengestützte Kommunikation sowie andere Infrastrukturnetze mit Adhoc Netzen, insbesondere mit MANETs kombiniert. Allerdings werden in diesem Projekt DTNs nicht betrachtet. Eine Kombination von MANETs mit DTNs bringt weitere Vorteile. Kommunikation ist mit Hilfe von DTNs auch mit abgespaltenen Netzteilen möglich.

Um auf einem Knoten einen verzögerungstoleranten Dienst (DTN) anzubieten, muss dieser im Vorfeld auf den Knoten implementiert sein. Nicht alle mobilen Geräte sind mit einem solchen Dienst ausgestattet. Die nicht DTN-fähigen Knoten können trotzdem zum Weiterleiten von Nachrichten genutzt werden, wenn diese direkten Kontakt zu anderen Knoten besitzen.

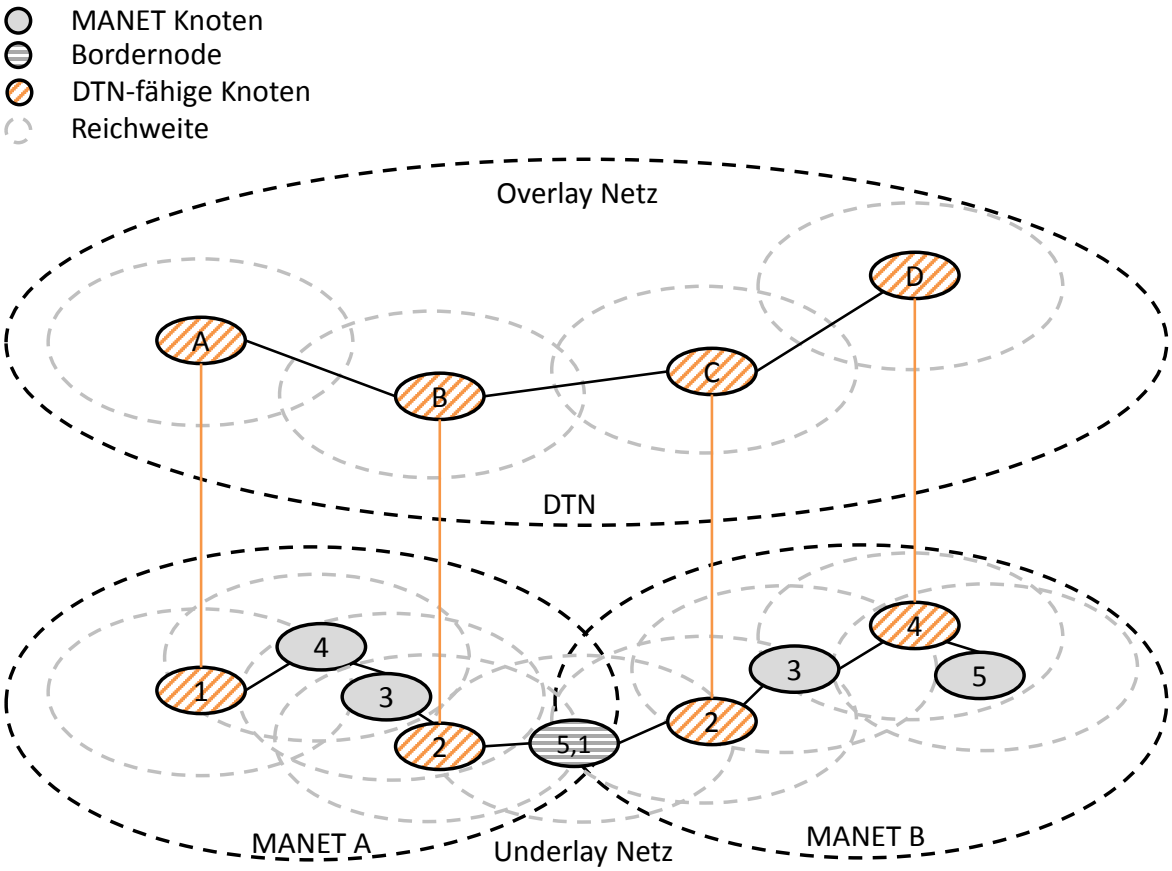

Abbildung 2.4: Overlay Netz

Abbildung 2.4 präsentiert ein DTN, welches als Grundlage zwei MANETs hat, in der nicht alle Knoten DTN-fähig sind. Dabei ist das MANET A mit MANET B über einen Bordernode (Definition 21) miteinander verbunden. In diesem Fall ist dieses DTN ein Overlay-Netz (Definition 22).

In Abbildung 2.4 ist zu erkennen, dass die Funkreichweite zwischen den DTN-Knoten nicht ausreicht, um einen direkten Kontakt herzustellen. Das Overlay-Netz bzw. DTN wird logisch mit Hilfe der MANET-Knoten verbunden. Dabei stellen die Protokolle der unterliegenden Netze eine Verbindung zu den Knoten her.

Definition 20 (Router) *Router haben die Aufgabe, Daten zwischen Computern in verschiedenen Netzen auf möglichst günstigen Wegen weiterzuleiten. Sie verbinden, laut International Organization for Standardization (ISO)/OSI-Referenzmodell (siehe Abschnitt 2.3.1) auf der Vermittlungsschicht (Schicht 3) auch Netze unterschiedlicher Topologien. [Plate, 2015]*

Dieses Beispiel zeigt, dass sich der Adressraum der jeweiligen Netze überschneidet. Gleiche Adressen führen zu Problemen bei der eindeutigen Identifizierung der Knoten. Die unterliegenden MANETs haben keine Kenntnis über die logischen Adressen (siehe Abschnitt 2.3.2) des darüber liegenden Overlay-Netzes. Sendet beispielsweise der DTN-Knoten A eine Nachricht an den DTN-Knoten C, der die Adresse Knoten 2 im MANET B besitzt, wird diese Nachricht im MANET A wahrscheinlich an die Adresse Knoten 2 in diesem MANET weitergeleitet. Die Veröffentlichungen [Krug u. a., 2014a] und [Schellenberg u. a., 2015] beschäftigen sich mit diesem Problem.

Definition 21 (Bordernode) *Bordernode wird ein Knoten genannt, der zwei oder mehr autonome Netze bzw. MANETs miteinander verbindet. Er hat die gleichen Aufgaben wie ein Router (Definition 20). Ein Bordernode ist demzufolge Mitglied dieser Netze, und kann deshalb auch mehrere Netzadressen besitzen. Der Datenverkehr in fremde Netze erfolgt dann ausschließlich über diesen Knoten.*

Definition 22 (Overlay-Netz) *Ein Overlay-Netz ist ein Netz, welches auf einem oder mehreren bestehenden Netzen, den Underlay Netzen, virtuell oder logisch aufgesetzt bzw. überlagert ist. Es versteckt unterschiedliche Netztypen und ermöglicht beispielsweise eine zusätzliche Adressierung. [Coulson u. a., 2005]*

2.3 Gruppenkommunikation

Kommunikation ist der Austausch von Informationen. Dieser Informationsaustausch kann anhand unterschiedlicher Eigenschaften eingeteilt werden. Auf Charakterisierungen nach Übertragungsverfahren oder Nutzungsrichtung wird in dieser Arbeit nicht eingegangen.

Wichtig hingegen ist die Einteilung anhand der Anzahl der beteiligten Partner. Dabei werden die Anzahl der Sender und Empfänger unterschieden. Die bekannteste Form ist die Unicastübertragung, bei der es genau einen Sender und einen Empfänger gibt (1:1- Kommunikation). Als Beispiel für diese Übertragungsart zählt das Herunterladen einer Datei oder die Telefonie. Beim Broadcast initiiert ein Sender eine Nachricht an alle Teilnehmer im Netz (1:Alle-Kommunikation), bekannt ist Radio und Fernsehen.

Der Notruf (Telefon 112), bei der sich die nächstgelegene Notrufzentrale meldet, ist ein Beispiel für die Anycast-Kommunikation (1:Erster-Kommunikaton), bei der die Nachricht an eine bestimmte Gruppe geht und der am nächsten gelegene Knoten dieser Gruppe antwortet. Die Übertragung von Daten von mehreren Sendern an nur einen Empfänger, wie beispielsweise die Erfassung von Sensordaten, wird als Concast (n:1-Kommunikation) bezeichnet. Die Kommunikation zwischen mehreren Sendern und Empfängern wird als Multipeer-Kommunikation (n:m-Kommunikation) angegeben und findet bei Videokonferenzen ihre Verwendung.

Die für diese Arbeit relevante Kommunikationsart ist die Gruppenkommunikation (Definition 23) und wird deshalb näher vorgestellt.

Definition 23 (Gruppenkommunikation) *Gruppenkommunikation, auch als Multicast bekannt, ist eine Kommunikationsform, in der ein Kommunikationsteilnehmer eine Nachricht an mehrere Empfänger (1:n) schickt. Dabei werden die Daten einmal versendet und Router erstellen Kopien und leiten diese bei Bedarf weiter. [Wittmann und Zitterbart, 2000]*

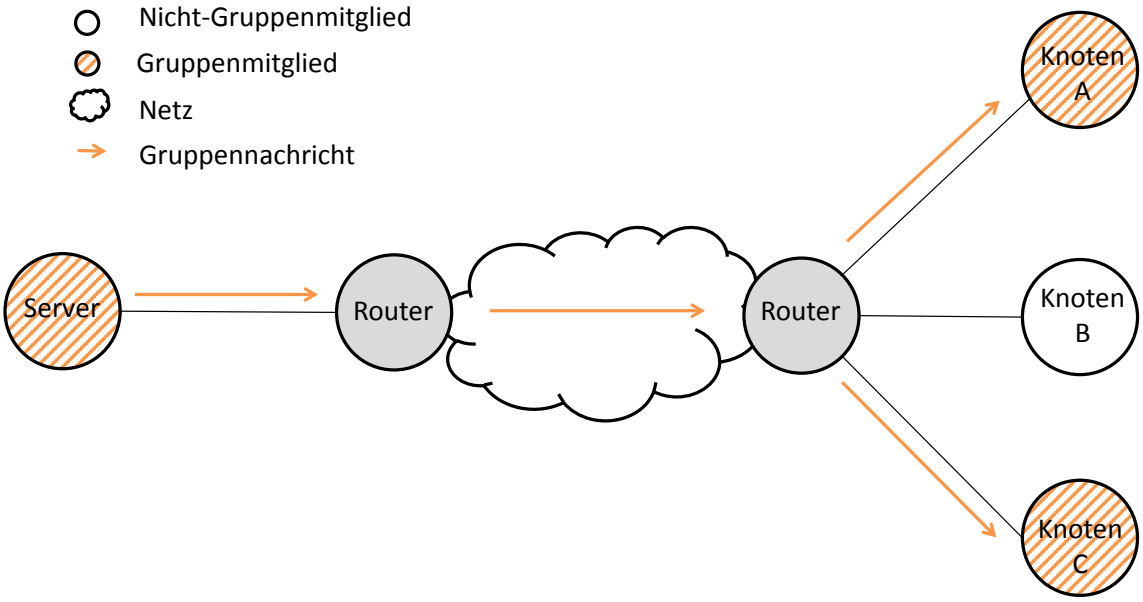

Abbildung 2.5: Gruppenkommunikation

Abbildung 2.5 zeigt eine schematische Darstellung der Gruppenkommunikation. Der Server auf der linken Seite sendet nur eine Nachricht durch das Netz zu verschieden Empfängern (Knoten A und Knoten C) auf der rechten Seite. Dabei wird die Nachricht, in diesem Beispiel, nur am letzten Router im Netz dupliziert und an die beiden Gruppenmitglieder zugestellt. Gruppenkommunikation vermeidet so redundanten Netzverkehr.

Eine Gruppenverwaltung bzw. ein Gruppenmanagement ist erforderlich, wenn dem Protokoll die Gruppenmitglieder bekannt sein müssen. Es existieren jedoch Ansätze, in denen kein Beitritt zu einer Gruppe, und demzufolge keine Gruppenverwaltung, erforderlich ist, denn dabei werden Gruppen anhand von Attributen identifiziert oder die Adresse von Zielknoten werden an die Gruppennachricht angehängt.

Bei der Gruppenkommunikation kann nach Art der Gruppen unterschieden werden. Hierbei wird, nach [Borghoff und Schlichter, 1995], zwischen offenen und geschlossenen Gruppen klassifiziert. In einer offenen Gruppe dürfen auch Knoten Gruppennachrichten versenden, welche nicht der Gruppe angehören. In einer geschlossenen Gruppe dürfen demzufolge nur Gruppenmitglieder Nachrichten an die Gruppe senden.

Ein weiteres Unterscheidungsmerkmal von Gruppen sind statische Gruppen und dynamische Gruppen. Bei einer statischen Gruppe ändern sich die Gruppenmitgliedschaften während der Existenz der Gruppe nicht. Bei einer dynamischen Gruppe können einzelne Gruppenmitglieder aus einer Gruppe austreten und neue Mitglieder einer Gruppe beitreten. Neue Gruppen können entstehen, während andere aufgelöst werden. Gruppen können überlappen.

Diese Form der Gruppenkommunikation setzt ein Gruppeverwaltung voraus. Folgende Verwaltungsfunktionen kennzeichnen im Allgemeinen die Gruppenkommunikation:

- eine Gruppe erstellen
- eine Gruppe bekanntmachen
- einer Gruppe beitreten
- eine Gruppe verlassen
- eine Gruppe löschen.

Die Geocast-Kommunikation ist eine Sonderform der Gruppenkommunikation. Dabei werden Nachrichten in ein räumlich abgegrenztes Gebiet gesendet.

2.3.1 Gruppenkommunikation in Referenzmodellen

Auch der Gruppenkommunikationsdienst kann in Referenzmodelle eingeordnet werden. Gruppenkommunikationsdienste werden entweder durch das Versenden von Unicastnachrichten an alle Gruppenmitglieder realisiert oder durch die Ausnutzung multicastfähiger Netze. Sie kann man auf den verschiedensten Schichten des ISO/OSI-Referenzmodell oder auch im Internet-Referenzmodell (Abbildung 2.6) finden.

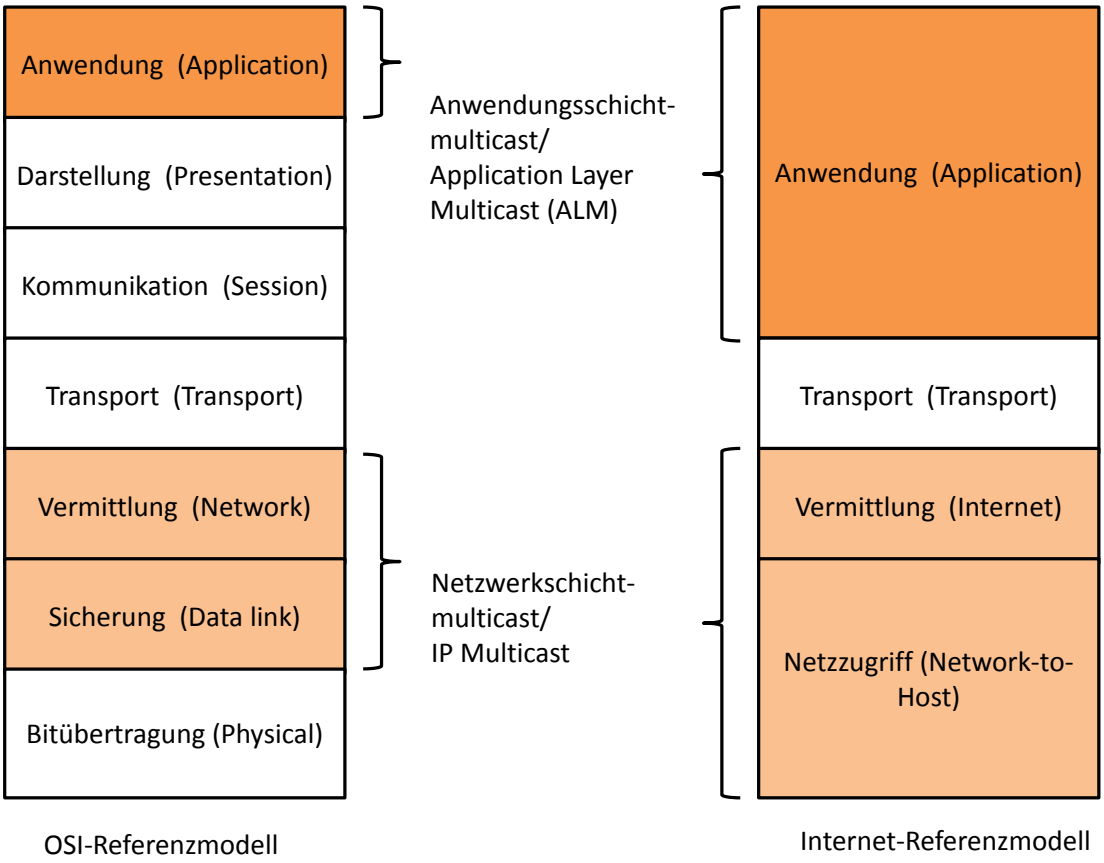

Abbildung 2.6: Schichtenmodelle Gruppenkommunikation

Die Gruppenkommunikation auf der Sicherungsschicht dient als Grundlage für die Gruppenkommunikation auf der Vermittlungsschicht und wird innerhalb eines LANs angewendet. In WANs befindet sie sich auf der Vermittlungsschicht. IP-Multicast ist das bekannteste Protokoll für die Gruppenkommunikation auf der Vermittlungsschicht. Dabei werden die Multicastadressen (Gruppenadressen) der Vermittlungsschicht auf Multicastadressen der Sicherungsschicht abgebildet.

IP-Multicast wird als Basisdienst für sichere Multicastübertragung auf der Transportschicht verwendet. Multicastfähige Transportprotokolle erweitern die Vermittlungsdienste z. B. um Reihenfolgeerhaltung oder Zuverlässigkeit.

Application Layer Multicast (ALM), auch Overlay Multicast genannt, ist die Gruppenkommunikation auf der Anwendungsebene. Die Teilnehmer bilden ein Overlay-Netz zum Datenaustausch. Auf dieser Ebene ist die Gruppenkommunikation unabhängig von der darunter liegenden Netztopologie, weshalb keine Anpassung von Netzkomponenten notwendig ist. Kennzeichnend für ALM ist, dass jede Gruppe durch einen eigenen Identifikator/Identifikation (ID) identifiziert wird.

2.3.2 Adressierung

Um eine Gruppe identifizieren zu können, benötigt eine Gruppe eine Gruppenadresse (Multicastadresse). Möchte ein Knoten eine Nachricht an eine Gruppe versenden, nutzt er diese Gruppenadresse als Zieladresse. Jede Schicht verwendet verschiedene Adressierungsarten. Deshalb ist es notwendig, die Gruppenadressen in dem jeweilig anderen Adressierungsschema abzubilden. Diese Umsetzung wird auch Mapping genannt. Folglich ist es nicht notwendig, dass die DTN-Schicht die unterliegende Gruppenadresse kennt. Ein bekanntes Protokoll für das Adress-Mapping ist das Address Resolution Protocol (ARP), welches IP-Adressen der jeweiligen Hardwareadresse zuordnet.

MANETs verwalten einen eigenen Adressraum. Beim Adress-Mapping werden oftmals Namen auf IP-Adressen umgesetzt. Ein Konzept zur konsistenten Namensauflösung mit adaptiven Routingtechniken stellen [Schellenberg u. a., 2013] vor. Es können Probleme bei der Namensauflösung sowie bei der Weiterleitung von Nachrichten entstehen, wenn MANETs den gleichen Adressraum verwenden und miteinander kommunizieren. Eine eindeutige Adressierung ist somit nicht gegeben, siehe dazu auch Abbildung 2.4 aus dem Unterkapitel 2.2. Eine Lösung für dieses Problem stellen [Krug u. a., 2014a] für Unicastnachrichten vor. Dieser Ansatz verwendet Zuordnungstabellen in den Bordernodes um eine eindeutige Adresszuordnung zu ermöglichen. Eine Erweiterung dieses Ansatzes für DTNs ist in [Schellenberg u. a., 2015] beschrieben. Die Umsetzung einer eindeutigen Multicastadressierung für mehrere MANETs oder DTNs ist viel umfassender und in der Literatur nicht beschrieben.

Um einen Überblick über die Komplexität der Gruppenadressierung zu bekommen, folgt in den folgenden Kapiteln eine ausführlichere Beschreibung der Gruppenadressierung auf den verschieden Schichten.

2.3.2.1 Anwendungsschichtmulticastadressierung

Nicht jedes Netz ist multicastfähig, weshalb die Gruppenkommunikation von der Anwendungsschicht übernommen werden kann. Die Adressierung erfolgt mittels eindeutiger ID. Diese ID ist völlig unabhängig von den darunter liegenden Schichten bzw. dem darunter liegenden Netz. Ein großer Vorteil ist, dass kein Wartungsaufwand im Netz notwendig ist. ALM ist immer gekoppelt mit einer Anwendung.

Die eindeutige ID kann durch unterliegende Adressierungsalgorithmen erstellt werden und beispielsweise via Uniform Resource Locator (URL)-Notation dargestellt werden. Eines der ersten ALM-Protokolle ist das Your Own Internet Distribution (YOID)-Protokoll [Francis, 1999]. „yoid://rendezvous.name:port/groupName“ ist eine Beispiel-URL aus [Francis, 1999] für Gruppen.

2.3.2.2 Netzwerkschichtmulticastadressierung

Für die Adressierung in der Vermittlungsschicht sind derzeit vorrangig zwei Protokolle im Einsatz: das ältere Internet Protokoll Version 4 (IPv4) und das neue Internet Protokoll Version 6 (IPv6). In den nächsten Jahren soll IPv6 IPv4 vollständig ersetzten. Beide Protokolle besitzen einen festgelegten Multicastadressraum.

IPv4-Multicastadressierung

Für die Adressierung von Multicastgruppen im IPv4-Protokoll ist ein fester Adressraum vorgesehen. Die Multicastadresse, früher die Klasse-D-Adresse, ist in Abbildung 2.7 (standardisiert im RFC 3171 [Albanna u. a., 2001]) dargestellt. IPv4-Multicastadressen haben keine Subnetzmaske.

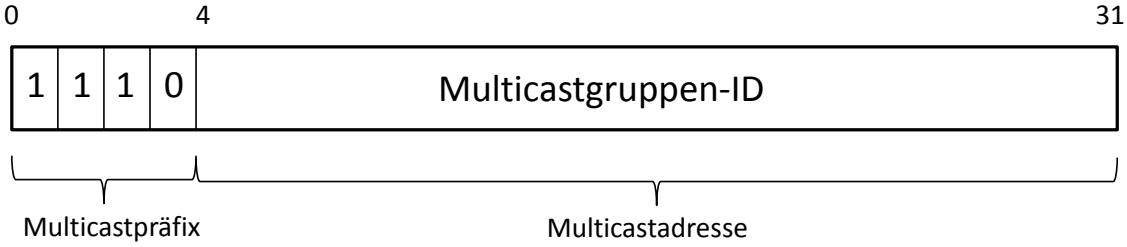

Abbildung 2.7: IPv4-Multicastadresse

Die Multicastadresse ist gekennzeichnet durch die ersten 4 Bits des ersten Oktetts 1110 (Multicastpräfix) und den 28 Bit Multicastadressteil. Sie befinden sich im Adressraum zwischen 224.0.0.0 bis 239.255.255.255. Die Quellenadresse eines Multicastpaketes ist immer eine Unicast-IPv4-Adresse.

IPv6-Multicastadressierung

IPv6-Multicastadressen beginnen immer mit 1111111b (Hexadezimal: FF). Anschließend werden 4 Bits für Flags reserviert, z. B. für vorübergehend zugewiesene Multicastadressen. Danach folgen weitere 4 Bits für den Gültigkeitsbereich, auch Scope genannt, z. B. steht 1110b (Hexamdezimal: E) für globalen Multicast. In Abbildung 2.8 ist die IPv6-Multicastadresse dargestellt, welche im RFC 2373 [Hinden und Deering, 1998] standardisiert ist.

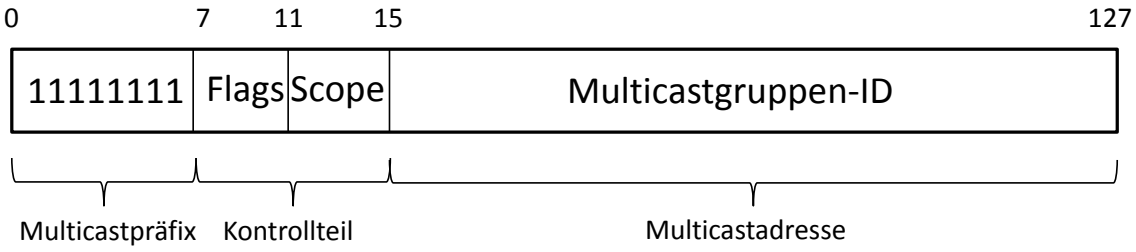

Abbildung 2.8: IPv6 -Multicastadresse

2.3.2.3 Sicherungsschichtmulticastadressierung

Die Adressierung in der Sicherungsschicht (Schicht 2) ist direkt abhängig von den IP-Adressen aus der Vermittlungsschicht. Beim Mapping werden IPv4- und IPv6-Multicastadressen auf Pseudo-Media-Access-Control (MAC)-Adressen abgebildet. [Wikipedia, 2015b] Am Beispiel von Ethernet soll das Mapping dargestellt werden. Ethernet-Gruppenadressen werden laut [Eastlake und Abley, 2013] (RFC 7402) mit 01:00:5E ge-

kennzeichnet. Der Multicastadressbereich beginnt bei 01:00:5E:00:00:00 und endet bei der Adresse 01:00:5E:7F:FF:FF. Die Hälfte des Blocks ist für die eigentliche Multicastadresse reserviert. In Abbildung 2.9 ist ein Beispiel dargestellt. In diesem Beispiel wird die MAC-Adresse 01-00-5e-0a-00-01 auf die IPv4-Multicastadresse 224.10.0.1 umgesetzt. Ersetzt man die mit x gekennzeichneten Bits mit allen Kombinationen von 0 und 1 erhält man den IPv4-Multicastadressbereich zwischen 224.10.0.1 und 239.138.0.1. Alle diese IPv4-Multicastadressen werden auf die MAC-Adresse 01-00-5e-0a-00-01 geleitet.

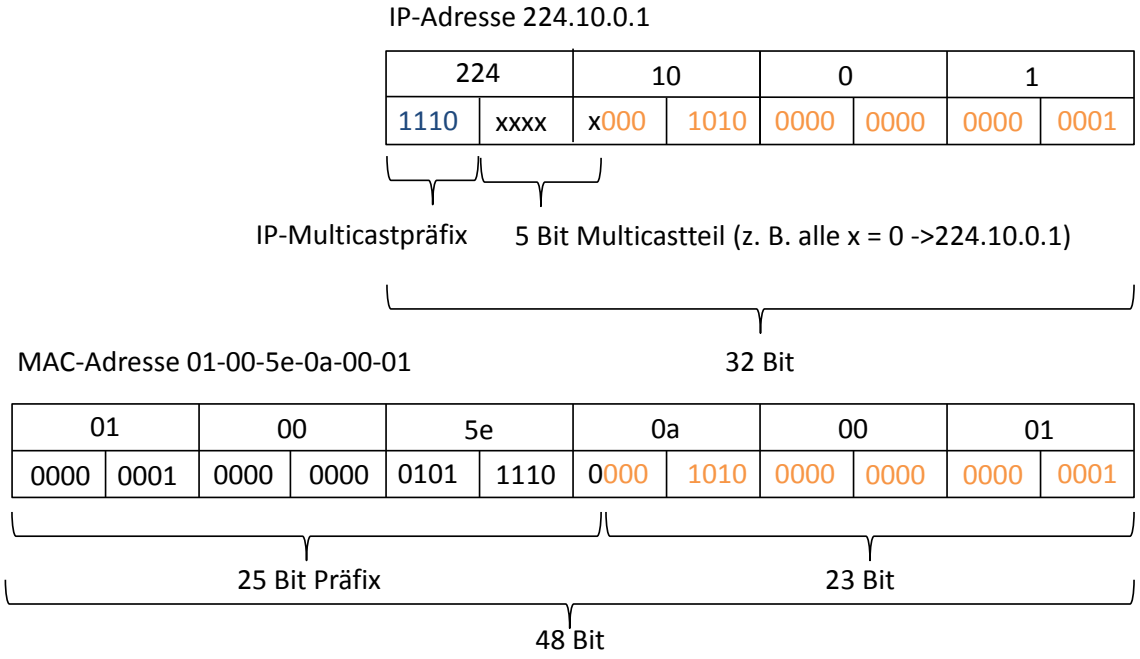

Abbildung 2.9: Umsetzung von IPv4-Multicastadressen auf MAC-Adresse

2.3.2.4 Geocast-Adressierung

Gruppenadressen beim Geocast kennzeichnen ein bestimmtes geografisches Gebiet. Dabei wird zwischen geometrischer Adressierung und symbolischer Adressierung unterschieden. [Bärwald, 2009]

Die geometrische Adressierung beschreibt einen Zielbereich mit absoluten Koordinaten. Diese werden entweder durch geometrische Figuren wie Rechtecke, Kreise oder Polygone abgebildet oder durch Satellitennavigationssysteme, wie z. B. durch das amerikanische Global Positioning System (GPS) oder zukünftig durch das sich in der Entwicklung befindende Satellitennavigationssystem Galileo.

Bei der symbolischen Adressierung wird das Zielgebiet mit einem Alias wie Zimmernummern, Straßennamen, Länder oder Städte gekennzeichnet. Die Kenntnis der Router über diese Aliasse ist Voraussetzung für die richtige Weiterleitung solcher Nachrichten. Symbolische Adressen sind intuitiv anzuwenden und leicht zu merken, müssen aber in eine geometrische Adressierung umgesetzt werden.

2.3.3 Zuverlässige Gruppenkommunikation

Traditionelle Aussagen über Zuverlässigkeit in der Gruppenkommunikation gehen davon aus, dass Daten bzw. Gruppennachrichten fehlerfrei, in der richtigen Reihenfolge und ohne Duplikate ausgeliefert werden. [Schiller und Boigt, 2003]

Hauptsächlich erfolgt eine Einteilung in Zuverlässigkeitsgrad, d. h. Garantien, die die Empfängeranzahl betreffen, oder eine Einteilung der Ordnung, d. h. die Empfangsreihenfolge betreffend.

Eine einheitliche Definition zuverlässiger Gruppenkommunikation ist nicht existent, was auch die nachstehenden Literaturbeispiele zeigen.

[Mankin u. a., 1998] stellten schon 1998 fest, dass Multicastanwendungen unterschiedliche Anforderungen an die Zuverlässigkeit stellen. Dies ist beispielsweise die Reihenfolgetreue der Daten. Bei Audio- oder Video-Übertragungen ist eine 100 % Zustellung der Datenpakete nicht gefordert, wohingegen die Synchronisation von Datenbanken eine zuverlässige Datenübertragung verlangt.

[Atwood, 2004] klassifizierte Zuverlässigkeit je nach Exaktheit der Datenlieferung von der Transportschicht an die höhere Schicht. In seiner Arbeit bezog er sich auch auf die Aussagen von [Handley u. a., 2000], welche die Anforderungen an die Zuverlässigkeit auch von der Gruppengröße abhängig machen.

Eine andere Unterteilung von zuverlässiger Multicastdatenübertragung präsentierte [Diot, 1995]. Dort wurde erkannt, dass es verschiedene Klassifizierungen von Zuverlässigkeit, je nach Anspruch der Anwendung, geben muss. Deshalb wurden dort verschiedene Klassen von Zuverlässigkeit definiert, beispielsweise „Verbindungsloser Multicast“ für einfaches Senden ohne Quittung oder „Ordnung“, bei der je nach ausgewählter Ordnungsregel Nachrichten in einer bestimmten Reihenfolge beim Empfänger ankommen müssen.

Für diese Arbeit wurde zuverlässige Multicastübertragung, wie in Definition 24 beschrieben, festgelegt.

Definition 24 (Zuverlässige Multicastübertragung) *„Zuverlässigkeit ist definiert als die erfolgreiche Übertragung von Gruppennachrichten an alle beabsichtigten Empfänger.“ [Begerow u. a., 2014a]*

Dabei ist zu beachten, dass eine Garantie der Zustellung an alle Empfänger in DTNs nicht gegeben werden kann, da, wie im folgenden Kapitel 3 erläutert, besondere Kriterien für DTNs gelten. Details zur Zuverlässigkeit werden im Unterkapitel 3.2 diskutiert.

3 Anforderungen und Bewertungskriterien für DTNs

Durch die besonderen Eigenschaften von DTNs, wie unterbrochene Verbindungen, große Verzögerungen, asynchrone Datenübertragung und beschränkte Ressourcen werden besondere Anforderung an das Routing sowie an die zuverlässige Datenübertragung gestellt. In diesem Kapitel werden Routingeigenschaften von DTNs eingeführt, eine Klassifizierung der Routingprotokolle vorgenommen sowie der aktuelle Stand der Technik vorgestellt. Anschließend folgt die Einführung des Begriffes Zuverlässigkeit in Bezug auf DTNs.

3.1 Routing in DTNs

Routing (Definition 25) spielt in Netzen eine wichtige Rolle. Routing, auch Wegewahl genannt, vermittelt Nachrichten bzw. Datenpakete vom Sender bis zum Empfänger über geeignete Wege. Dabei können bei Wegewahlentscheidungen die Kosten, beispielsweise die Leitungskapazität, die Leitungskosten oder die Anzahl der Hops, mit einbezogen werden. Nicht jedes Routingprotokoll (Definition 26) ist für jeden Netztyp geeignet.

Definition 25 (Routing) *„Unter Routing versteht man eine Wegwahlfunktion zur Vermittlung von Nachrichten zwischen LANs, zwischen LANs und WANs sowie zwischen WANs. Im Gegensatz zum Bridging, das ebenfalls zur Datenflusssteuerung benutzt wird und auf der Sicherungsschicht erfolgt, findet das Routing auf der Vermittlungsschicht statt.“ [Lipinski u. a., 2015f]*

Definition 26 (Routingprotokoll) *„Routingprotokolle sind Protokolle mit denen die Router untereinander kommunizieren. Sie dienen dazu, die Wegwahl für die Vermittlung von Nachrichten über mehrere Netze hinweg zu optimieren. Die optimale Wegwahl kann kosten- oder bandbreitenoptimiert sein, sie kann die Auslastung der Verbindung berücksichtigen, die Anzahl der Hops, die Übertragungsgeschwindigkeit oder das Echtzeitverhalten.“ [Lipinski u. a., 2015g]*

Routingprotokolle entwickelt für MANETs sind im Gegensatz zu den Routingprotokollen für das Internet auf die Eigenschaften von MANETs angepasst. Routingprotokolle für DTNs wurden wiederum auf dessen Eigenschaften abgestimmt. Deshalb werden im folgendem Abschnitt 3.1.1 allgemeine Routinganforderungen für DTNs dargelegt und näher erläutert.

3.1.1 Allgemeine Routingeanforderungen in DTNs

Um optimale Routingergebnisse zu erreichen wurden verschiedene Anforderungen für DTNs identifiziert. [Mehta und Shah, 2014] definieren fünf Anforderungen, die beim Routing in DTNs bedachtet werden sollten:

- Speicherkapazität
- Energie
- Zuverlässigkeit
- Prozessorleistung
- Sicherheit.

Speicherkapazität

Ein wichtiger Aspekt ist die Speicherkapazität der einzelnen Knoten in DTNs. Der Speicher der Knoten ist für das Zwischenspeichern der Nachrichten bis zu ihrer möglichen Weiterleitung notwendig. Normale Netzknoten haben dabei oft geringere Speicherkapazität, als z. B. Fähren, die zum Transport und Weiterleiten von Nachrichten konfiguriert sind.

Energie

Der Einsatz mobiler Geräte ist abhängig von der Kapazität ihrer Batterien. Neben der Energie, die durch Fortbewegung eines Knotens verbraucht wird, wird während des Empfangens, Speicherns und Weiterleitens von Nachrichten auch Energie benötigt. Diesen Energieverbrauch gilt es so gering wie möglich zu halten. [Mehta und Shah, 2014]

Zuverlässigkeit

Zuverlässige Datenübertragung steigert die Qualität von Routingprotokollen. Das Wissen, dass der Empfänger die Nachricht erhalten hat, ist oft essentiell für den Sender. Quittungen sind deshalb eine Methode, um den erfolgreichen Nachrichtenaustausch zu bestätigen. [Mehta und Shah, 2014]

Prozessorleistung

Die Prozessorleistung in mobilen Geräten hat sich in den letzten Jahren verbessert. Trotzdem ist der Rechenaufwand für solche Geräte möglichst gering zu halten. Komplexe Routingprotokolle mit hohem Rechenaufwand sind deshalb für DTNs eher ungeeignet.

Sicherheit

Die Sicherheit ist in jedem Netztyp zu beachten. Katastropheneinsätze bedürfen besonderer Maßnahmen um Missbrauch oder Falschmeldungen zu verhindern. Nachrichten werden oftmals über mehrere Knoten versendet, bis diese ihre endgültigen Ziele erreichen. Authentifizierung und Verschlüsselungstechniken sollten aus Sicherheitsgründen in die Überlegung beim Entwickeln von Routingprotokollen einbezogen werden.

3.1.2 Klassifizierung von Multicastroutingstrategien

In diesem Kapitel werden allgemeine Multicastroutingstrategien für DTNs vorgestellt. Diese Routingstrategien haben Einfluss auf die zuverlässige Auslieferung von Gruppennachrichten. Je nach Anwendungsfall sind diese Strategien besser oder weniger gut geeignet. Bestehende Ansätze, werden im Unterkapitel 3.3 aufgezeigt und diskutiert.

Nach [Özcan u. a., 2011] lassen sich DTN-Routingprotokolle in vier Hauptkategorien unterteilen:

- Flutenbasiertes Routing
- Selektives Routing
- Wahrscheinlichkeitsbasiertes Routing
- Intelligentes Routing.

Eine allgemeine Bewertung dieser Routingansätze ist in Tabelle 3.1 dargestellt. Diese Kategorien werden unter anderem anhand des Speicherbedarfs, der Auslieferungsrate bzw. Zuverlässigkeit und dem Overhead (Definition: 27) bewertet.

Tabelle 3.1: Allgemeine Routingkategorien in DTNs

Kategorie	Fluten-basiertes Routing	Selektives Routing	Wahrschein-lichkeits-basiertes Routing	Intelligentes Routing
Routingart	flutenbasiert	baumbasiert	basiert auf Kontaktein-formationen	hybrid oder fährenbasiert
Speicherbe-legung	sehr hoch	gering	mittel	hoch bei Fäh-ren, sonst ge-ring
Auslieferungs-rate	sehr hoch	gering	hoch	hoch
Netzeignung	geeignet für hoch mobile Netze	nur geeignet für relativ statische Netze	für hoch mobile Netze mit konstan-tem Bewe-gungsmuster geeignet	bedingt für hoch mo-bile Netze geeignet
Overhead	sehr hoch	mittel	mittel	gering
Verzögerung	gering	hoch	mittel	mittel bis sehr hoch bei Fähren

3.1.2.1 Flutenbasiertes Routing

Im flutenbasierten oder auch multicopy Routing werden Gruppennachrichten durch das Netz geflutet. Jeder Knoten speichert eine Kopie der Nachricht und gibt diese an andere Knoten weiter. Das Netz wird dadurch sehr belastet, was zu einem hohen Overhead (Definitition 27) führt. Durch die Speicherung der Nachricht in quasi jedem Knoten wird beim Ausfall eines oder mehrere Knoten dennoch eine hohe Auslieferungsrate erreicht. Vorteilhaft ist hierbei die geringere Verzögerungszeit. Dieser Routingansatz ist auch für hoch mobile Netze geeignet. Nachteilig ist aber, dass der Nachrichtenspeicher in den einzelnen Knoten sehr schnell an seine Grenzen gerät.

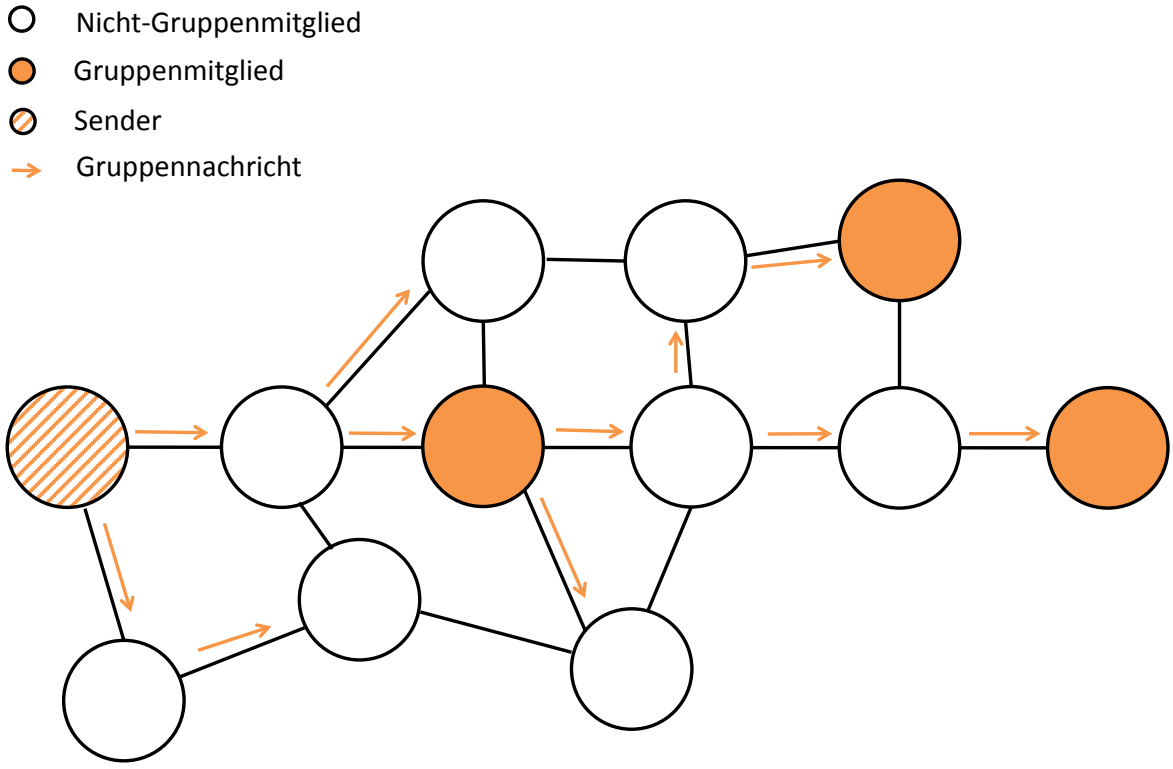

Abbildung 3.1: Flutenbasiertes Routing

Definition 27 (Overhead) *Als Overhead werden alle Informationen, wie Routinginformationen, Kontrolldaten und duplizierte Nutzdaten, die zusätzlich zu den eigentlichen Nutzdaten bzw. der Nachricht übertragen werden, bezeichnet. Diese Daten sind, beispielsweise für die Fehlerkorrektur, technisch erforderlich. [Lipinski u. a., 2015e]*

Abbildung 3.1 zeigt ein Beispiel, wie die Nachricht an alle Knoten weitergereicht wird, die die Nachricht noch nicht gespeichert haben. Das bekannteste Protokoll ist das Epidemic-Protokoll [Vahdat und Becker, 2000].

3.1.2.2 Selektives Routing

Im Gegensatz zum flutenbasierenden Routing werden Nachrichten entlang eines Multicastbaumes weitergeleitet. Ein Multicastbaum enthält den Pfad vom Quellknoten bis zu allen Zielknoten respektive Gruppenmitgliedern. Die Gruppennachricht wird an den Stellen dupliziert, an denen sich der Pfad auf verschiedene Knoten aufteilt (Abbildung 3.2).

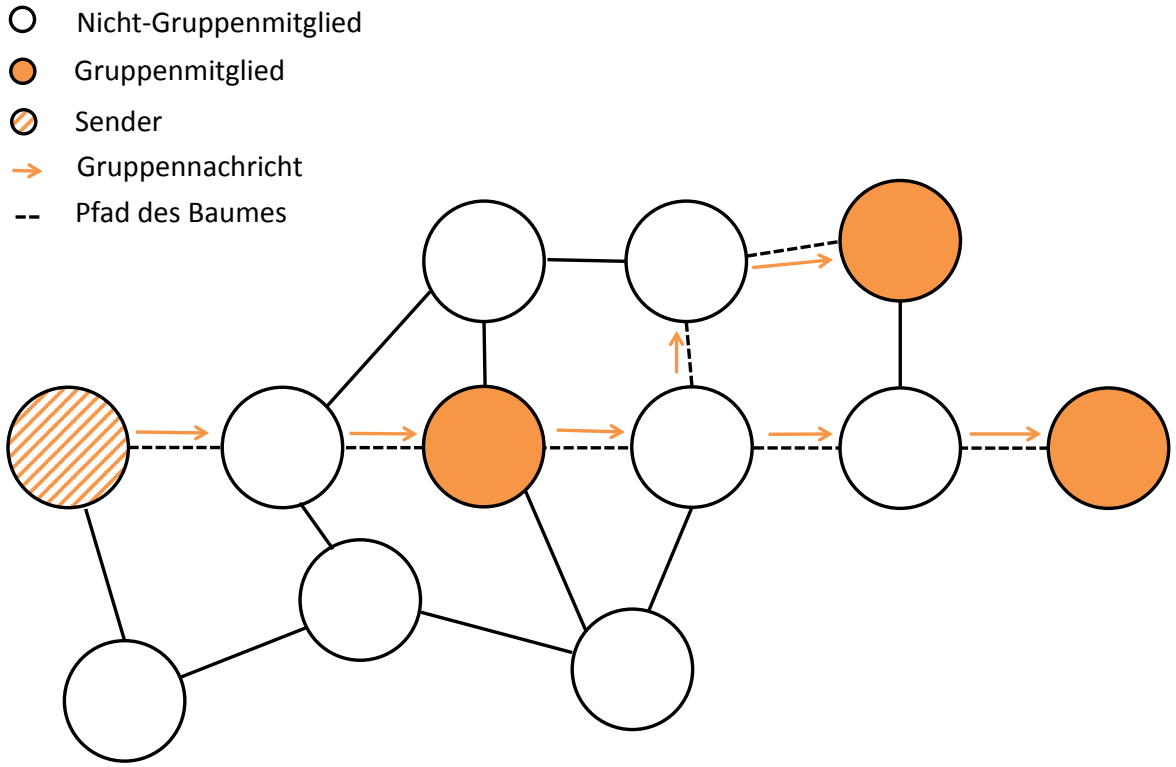

Abbildung 3.2: Selektives Routing

Protokolle dieser Kategorie besitzen eine geringe Speicherbelegung, da sich nur eine oder wenige Kopien im Netz befinden. Dies kann allerdings schnell zum Verlust einer Nachricht führen, falls ein Knoten ausfällt. Dann wird nur noch eine niedrige Zustellrate erreicht, was die Zuverlässigkeit in hoch mobilen Netzen beeinflusst. Oft entstehen hohe Verzögerungen bei der Weiterleitung, da der Knoten, zu dem die Nachricht weitergeleitet werden soll, gerade nicht in Reichweite ist. Multicastbäume sind

im Allgemeinen für hoch mobile Netze nicht geeignet. Nachrichten werden nur entlang des Multicastbaumes gespeichert, weshalb kaum Overhead entsteht. Trotzdem schneidet selektives Routing im Bereich Overhead nur mittelmäßig ab, da ein zusätzlicher Informationsaustausch erfolgen muss, um die Baumstruktur aktuell zu halten.

Ein großer Nachteil ist die Verzögerung der Auslieferung von Nachrichten, welche entsteht, wenn ein Knoten ausfällt und der Multicastbaum neu aufgebaut wird. Dynamic-Tree-Based Routing (DTBR) [Zhao u. a., 2005] ist ein Protokoll dieser Kategorie.

3.1.2.3 Wahrscheinlichkeitsbasiertes Routing

Gegenüber den beiden vorigen Routingansätzen wird hier eine Statistik zugrunde gelegt. Jeder Knoten merkt sich, wann er welchen Knoten zuletzt kontaktiert hat. Anhand dieses Wissens wird die Nachricht dann weitergeleitet, wenn ein anderer Knoten eine höhere Wahrscheinlichkeit hat, diese Nachricht auszuliefern (siehe Abbildung 3.3).

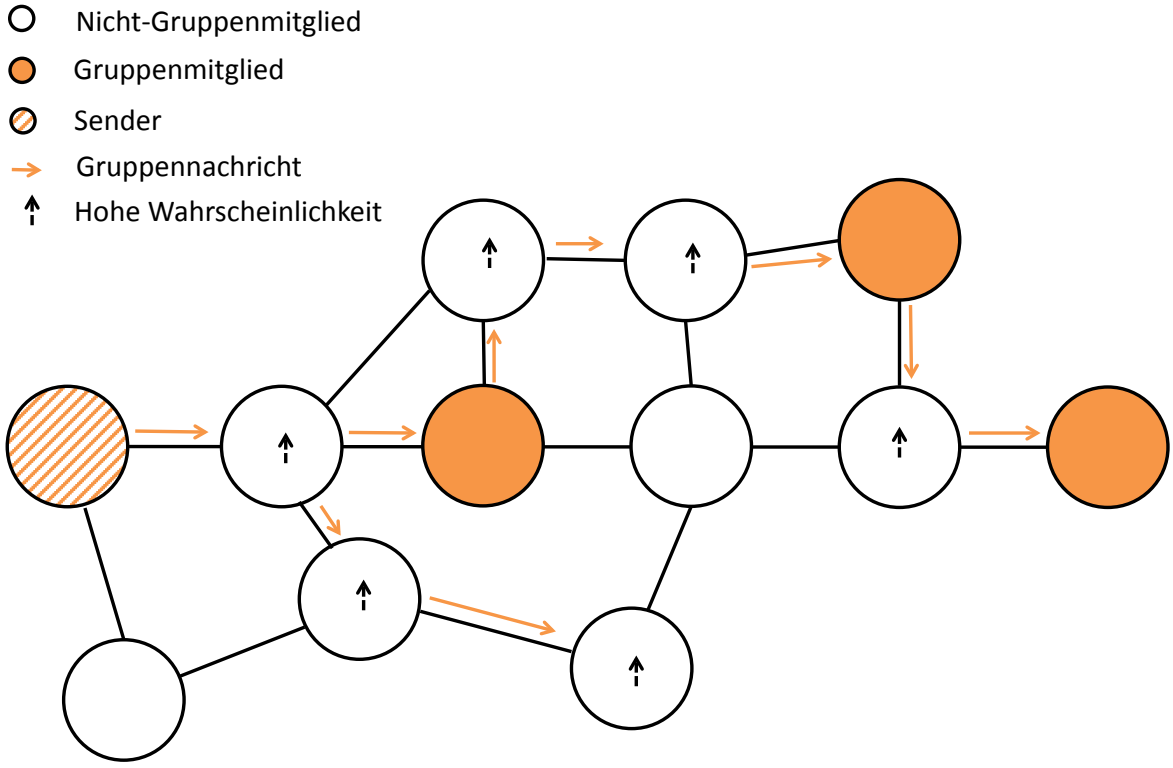

Abbildung 3.3: Wahrscheinlichkeitsbasiertes Routing

Dadurch wird die Anzahl der Kopien im Netz minimiert. Das führt zu einer geringeren Speicherauslastung und zu einem geringeren Overhead. Allerdings müssen die Statistikdaten gespeichert und ausgewertet werden. Die Auslieferungsrate ist hoch. Als Beispiel dafür steht das in [Lindgren u. a., 2012] standardisierte Protokoll Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET). Ein weiteres Protokoll dieser Familie ist das Adaptive Reinforcement-Based Routing (ARBR) Protokoll [Elwhishi u. a., 2010], welches zusätzlich die Kosten betrachtet. Das Bewegungsmuster sollte bei diesem Routingverfahren relativ konstant sein. Ist das der Fall, kann es auch für hoch mobile Netze eingesetzt werden. Zeigen die Knoten plötzlich gegensätzliche Bewegungsmuster, gehen Nachrichten verloren. Die Berechnung der Wahrscheinlichkeit, einen Knoten zu treffen, wird noch nach dem alten Bewegungsmuster erstellt. Das birgt die Gefahr in sich, dass die Nachricht an einen Knoten weitergereicht werden soll, welcher sich jedoch nach aktuellem Bewegungsmuster gegensätzlich bewegt.

3.1.2.4 Intelligentes Routing

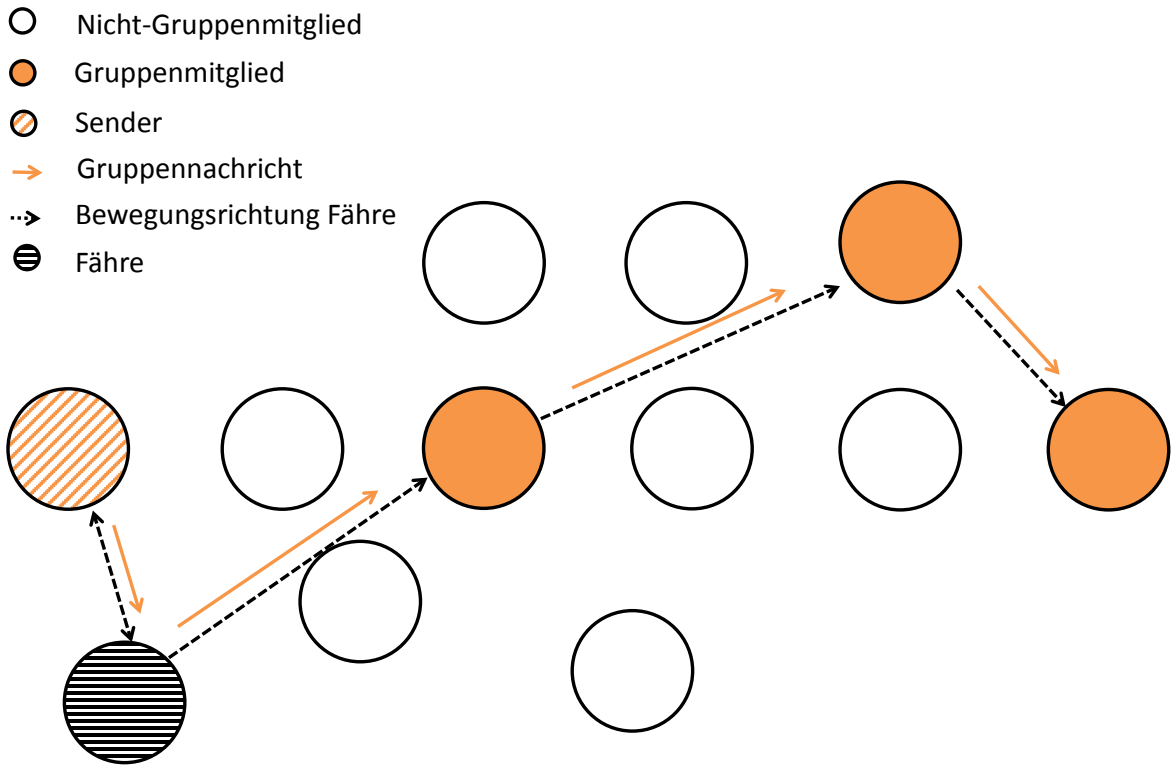

Abbildung 3.4: Intelligentes Routing mit Fähren

Die meisten Protokolle, die intelligentes Routing verwenden, nutzen einen hybriden Ansatz, beispielsweise eine Kombination von flutenbasiertem und selektivem Routing oder auch komplett andere Ansätze wie den Ameisenalgorithmus. Als Beispiel steht hier das Cultural Greedy Ant (CGrAnt) Protokoll [Vendramin u. a., 2012].

Mit intelligenten Algorithmen ist grundsätzlich eine hohe Auslieferungsrate möglich. Außerdem kann dadurch der Speicherbedarf gesenkt werden. Die Verzögerung bei der Auslieferung ist relativ hoch. Der Overhead kann je nach Algorithmus stark variieren. Auch ist nicht jedes Protokoll dieser Kategorie für hoch mobile Netze geeinet.

Es gibt aber auch andere Ansätze, die spezielle Knoten als Nachrichtenübermittler (auch Fähren) deklarieren. Diese Knoten besitzen oft besonders große Speicherkapazität. Sie nehmen Nachrichten von Knoten entgegen, und nur diese Knoten liefern die Nachrichten über den direkten Kontakt mit den Zielknoten aus. Dies kann zu sehr hohen Verzögerungszeiten führen. Geht die Fähre verloren, sind demzufolge auch alle Nachrichten verloren. Die Zuverlässigkeit ist stark abhängig von der Robustheit der Fähren. Erreicht die Fähre ihre Ziel, ist die Auslieferungsrate respektive Zuverlässigkeit sehr hoch. Durch die Bündelung der Nachrichten auf nur einem Knoten wird die Auslieferungsrate nur als hoch eingestuft. Ein Beipiel ist in Abbildung 3.4 dargestellt. Diese Knoten sind oft Fahrzeuge, Flugobjekte wie z. B. Quadrokopter oder sogar Tiere. Hier wird zwischen zwei Klassen unterschieden, den Node Initiated Message Ferrying (NIMF)-Protokollen (Bsp.: [Yang u. a., 2005]) bei denen die Knoten wissen, wann und wo eine Fähre vorbei kommt, und den Ferry Initiated Message Ferrying (FIMF)-Protokollen (Bsp.: [Shah u. a., 2003]), bei denen die Fähre die Nachrichten bei den Knoten abholt.

3.2 Zuverlässigkeit in DTNs

Einer der wesentlichsten Aspekte bei der Kommunikation mit Hilfe eines verzögerungstoleranten Dienstes ist die Definition, mit welchen Metriken die Zuverlässigkeit gemessen wird. Verschiedene Ausarbeitungen, wie [Lo und Luo, 2012], [Li u. a., 2013] und [Antunes und Morla, 2009], beziehen sich hauptsächlich auf die folgenden drei Metriken.

1. *Auslieferungsrate*: Ist der Wert der tatsächlich gelieferten Nachrichten im Verhältnis zu den zu liefernden Nachrichten zu allen gewünschten Empfängern. [Lo und Luo, 2012]

2. *Verzögerung*: Ist eine Angabe der durchschnittlichen Ende-zu-Ende Verzögerung der gelieferten Nachrichten von der Quelle zu den Zielen. [Lo und Luo, 2012]

3. *Übertragungskosten*: Ist das Verhältnis aller übertragenen bzw. kopierten Nachrichten zu der Anzahl der gelieferten Nachrichten. [Lo und Luo, 2012]

Laut Definition von Zuverlässigkeit aus Abschnitt 2.3.3 für diese Arbeit, bei der eine zuverlässige Übertragung dann erfolgt ist, wenn alle Gruppenmitglieder die Nachricht erhalten haben, wurden drei Bedingungen identifiziert, um einhundert Prozent Zuverlässigkeit zu erreichen: [Begerow u. a., 2014a]

1. Das Multicastprotokoll muss nach einem Multicopy-Modell arbeiten. Beim Multicopy-Modell werden mehrere Kopien der Nachricht im Netz von verschiedenen Knoten gespeichert und weitergeleitet. Dieses Modell erreicht die höchste Auslieferungsrate als auch geringe Verzögerungszeiten.

2. Der Nachrichtenspeicher der einzelnen Knoten muss eine „unendliche“ Speicherkapazität besitzen, also soviel Speicherkapazität, dass keine Nachricht vor dem Zustellen gelöscht werden muss.

3. Multicastnachrichten dürfen keinerlei Zeiteinschränkungen haben.

Diese Bedingungen können in DTNs insbesondere in Katastrophenszenarien nicht erfüllt werden. Ein Multicopyroutingprotokoll ist realisierbar. Nicht alle Knoten verfügen über genug Speicherkapazität. Deshalb ist eine effektive Speicherveraltung sinnvoll. Nachrichten, insbesondere Warnungen, müssen schnellstmöglich zugestellt werden. Deshalb ist der dritte Punkt nicht durchführbar.

3.3 Stand der Technik in DTNs

In Deutschland wird bei den BOSs weitestgehend das digitale Funksystem TETRA verwendet. Teile der Feuerwehr bzw. der Rettungsdienste verwenden noch die analoge Funktechnik. Da die Bandbreite nur für Sprachnachrichten sowie Kurznachrichten

ausgelegt ist, ist TETRA für die Übertragung von Videodaten nicht geeignet. TETRA basiert auf einem Infrastrukturnetz. Fällt die Infrastruktur aus, ist TETRA im Direct Mode Operation (DMO)-Modus nutzbar. Im DMO-Modus können Geräte direkt miteinander kommunizieren. Die Nachrichten können nur durch unmittelbaren Kontakt zum Sender, zu einem Repeater (Definition 28) oder zu einem Gateway (Definition 29) empfangen werden. Diese Nachrichten werden nicht zu anderen Knoten weitergeleitet, wie beispielsweise in MANETs.

Im Katastrophenfall kann die Infrastruktur beschädigt sein und die Kommunikation über den DMO-Modus ist nur nutzbar bei direktem Kontakt zum Sender. In diesem Fall könnten auch Einsatzkräfte nicht kommunizieren und koordiniert werden. Dieses Defizit im TETRA-Standard wurde zum Anlass genommen um die Forschung auf dem Gebiet der Kommunikation in Katasrophenfällen verstärkt voranzutreiben. MANETs und DTNs wurden dabei besonders beleuchtet.

Definition 28 (Repeater) *„Ein Repeater ist eine Kopplungseinheit auf der ersten Schicht (Bitübertragungsschicht, physical layer) des OSI-Modells. Es findet nur eine Verstärkung aller eingehender Signale einschließlich von Störungen statt. Repeater sind keine eigene Stationen oder Knoten und erhalten keine eigene Adresse. Sie verhalten sich transparent zu den angeschlossenen Stationen.“ [Eicker, 2008]*

Definition 29 (Gateway) *Gateways werden Kopplungseinheiten zwischen Netzen genannt. „Dazu werden alle Systeme gezählt, die mehr als die ersten drei Schichten des OSI-Modells behandeln können. Grundsätzlich ist es ein System, das an der Verbindungsstelle zwischen zwei topologisch und technisch völlig unterschiedlichen Netzwerken steht. Es übernimmt alle notwendigen Anpassungen, z. B. der Pakete und Zeichensätze, an die jeweils benötigten Protokolle des anderen Netzwerkes.“ [Eicker, 2011]*

Protokolle, die auf reinen Ad-hoc Netzen basieren, sind zwar nicht abhängig von der Infrastruktur und die Knoten sind in der Lage zu kommunizieren, decken aber oftmals nur ein relativ kleines geographisches Gebiet ab, was bei größeren Katastrophen zu mehreren isolierten kleinen Netzen führt. Somit ist eine durchgehende Kommunikation zwischen der Einsatzleitung und Einsatzkräften bzw. untereinander so nicht möglich.

Wie schon erwähnt, sind klassische Routingprotokolle, die für das Internet oder andere Netztypen wie MANETs entwickelt worden sind, nicht geeignet für DTNs. Deshalb

wurde in den letzten Jahren verstärkt an Protokollen für DTNs gearbeitet. Dabei sind eine Vielzahl neuer Unicastprotokolle sowie einige Multicastprotokolle entstanden. Der Fokus dieser Arbeit liegt auf der Multicastübertragung bzw. Gruppenkommunikation, weshalb in den folgenden Kapiteln nur auf Multicastprotokolle eingegangen wird. Nicht nur das Routing selbst sondern auch die Gruppenverwaltung ist Grundlage für viele Routingprotokolle. Ein kurzer Überblick über existierende Ansätze wird in den folgenden Kapiteln dargestellt.

3.3.1 Gruppenverwaltung im DTN

Viele Multicastprotokolle setzen ein Gruppenmanagement bzw. eine Gruppenverwaltung voraus. Es existieren nur wenige Ansätze für das Gruppenmanagement in DTNs. Schon die DTNRG deklarierte in seiner Ausarbeitung über Multicast in DTNs [S. Symington, 2006] das Gruppenmanagement als offenen Punkt.

In [Zhao u. a., 2005] wurde erstmals das Gruppenmanagement angerissen. Ein Knoten sendet eine *JOIN request*-Nachricht an den DTN-Routingagenten für den Gruppenbeitrittswunsch und eine *LEAVE request*-Nachricht für den Austrittswunsch. Routingagenten in DTNs authentifizieren Knoten, übernehmen aber auch Gruppenmanagementaufgaben. Eine detaillierte Beschreibung des Gruppenmanagements ist leider nicht vorhanden.

Das 2010 vorgestellte Protokoll „MembersOnly“ [Nelson und Kravets, 2010] ist ein dezentrales Gruppenmitgliedsverwaltungsprotokoll. Dieses Protokoll teilt das Gruppenmanagement in vier Phasen ein. Der erste Schritt ist das Erstellen der Gruppe anhand von Attributen oder Rollen. Dabei geht das Protokoll davon aus, dass sich danach an der Gruppenmitgliedschaft nichts mehr ändert. Im zweiten Schritt wird die Gruppenliste durch das Netz geflutet. Jeder Knoten sammelt in Schritt drei diese Liste von verschiedenen Knoten und vergleicht diese mit den schon erhaltenen Listen. Dabei kann erkannt werden, ob ein nicht autorisierter Knoten der Gruppe beigetreten ist. Als letzten Schritt können diese Gruppeninformationen für das Routing bereitgestellt werden. Dieses Protokoll nutzt also die verschiedenen Listen um unautorisierte Knoten zu erkennen. Ein regulär später der Gruppe beigetretener Knoten könnte als Sicherheitsproblem fehlinterpretiert werden. Grundsätzlich ist der Austausch von Listen ein guter Ansatz und wird deshalb als Inspiration für das Gruppenmanagement dieser Arbeit angesehen.

3.3.2 Multicast Routing im DTN

Wie schon angeführt, werden durch die fehlende Ende-zu-Ende-Verbindung in verzögerungstoleranten Netzen besondere Routingalgorithmen benötigt. Nachrichten werden gespeichert und transportiert. Die im Abschnitt 3.1.1 definierten Routinganforderungen werden als Grundlage für die Bewertung herangezogen. Besondere Beachtung wird dabei dem Speichermanagement und der Zuverlässigkeit geschenkt. Weiterhin werden diese Protokolle in die, im Abschnitt 3.1.2 vorgestellten Klassifizierungen eingeordnet. Am Ende zeigt Tabelle 3.2 eine Übersicht der vorgestellten Multicastprotokolle.

[Zhao u. a., 2005] erkannten schon frühzeitig die Problematik von Multicast in DTNs. Es wurden vier verschiedene Strategien für Multicastrouting entwickelt und miteinander verglichen. Darunter sind selektive, wie Unicast-Based Routing (UBR), Static-Tree-Based Routing (STBR) und Dynamic-Tree-Based Routing (DTBR), sowie flutenbasierten Verfahren wie Group-Based Routing (GBR) und Broadcast-Based Routing (BBR). Bei UBR wird die eigentliche Multicastnachricht verpackt und als eigenständige Nachricht an jedes einzelne Gruppenmitglied adressiert. Der Vergleich ergab, dass flutenbasierte Strategien einen höheren Auslieferungsgrad haben als selektive Verfahren. Über Speicherverbrauch wurden keine Angaben gemacht. Das Beitreten bzw. Austreten aus bzw. in eine Gruppe erfolgt mittels Routingagent. In dieser Veröffentlichung wurden drei Modelle vorgestellt, welche zeitweise Einschränkungen der Gruppenmitgliedschaften betrachten sowie den Zeitpunkt der Nachrichtenauslieferung mit einbeziehen. Abhängig von der ausgewählten Option werden zusätzliche Informationen an alle Multicastnachrichten angefügt. Möglich sind der Gruppenmitgliedszeitraum, das Übermittlungsintervall und das Current-Member Delivery (CMD)-Flag. Infolgedessen wird die Gruppennachricht größer und es werden mehr Speicherressourcen benötigt. [Zhao u. a., 2005] gehen des Weiteren auf die unterschiedlichen Anforderungen bei der Auswahl der richtigen Empfänger innerhalb einer Gruppe ein. Es wird somit zur Basis für die im Abschnitt 4.4.2 beschriebene Empfängeridentifikation. In [Zhao u. a., 2005] wurde der Grundstein für die Gruppenkommunikation in DTNs gelegt. Hier wurde erkannt, dass Multicastrouting in DTNs, je nach Szenario, verschiedene Routingstrategien benötigt. Es werden aber keine Aussagen über die Zuverlässigkeit getroffen. In Katastrophenszenarien ist dies allerdings ein entscheidendes Kriterium. Auch hier ist die aktuelle Anzahl der Gruppenmitglieder nicht bekannt.

On-demand Situation-aware Multicasting (OS-Multicast) [Ye u. a., 2006] ist ein weiteres Beispiel für selektives Routing. Es passt den Multicastbaum entsprechend der

aktuellen lokalen Sicht an jeden Zwischenknoten an. Trotz alledem ist es für hoch dynamische DTNs ungeeignet. OS-Multicast beinhaltet einen Mechanismus zum früheren Löschen der Nachrichten. Dieses Protokoll verwaltet zwei Listen, die an jede Nachricht angehängt werden. Die erste Liste enthält die Knoten, an die die Nachricht weitergeleitet werden soll, und eine weitere Liste mit den Zielknoten. Sobald ein Knoten die Nachricht an alle gewünschten Empfänger über die Zwischenknoten weitergeleitet hat, kann die Nachricht gelöscht werden. Die Liste der erreichbaren Zielknoten wird an jedem Zwischenknoten aktualisiert. Das Hauptproblem dieses Ansatzes ist, dass alle Empfänger dem Sender bekannt sein müssen. Knoten, welche später der Gruppe beitreten, werden in diesem Protokoll nicht berücksichtigt. Bei OS-Multicast ist entlang des Baumes der Nachrichtenspeicher schnell ausgelastet. Die Verzögerung ist nur gering, wenn eine quasi Ende-zu-Ende Verbindung besteht, was in hoch mobilen DTNs nicht der Fall ist. Durch erhöhte redundante Nachrichtenspeicherung, verursacht durch mehrere verfügbare Links, wird mehr Overhead erzeugt und führt zu einer höheren Speicherauslastung. Nach [Ye u. a., 2009] hat OS-Multicast einen höheren Auslieferungsgrad und eine geringere Verzögerung als UBR, STBR und DTBR.

Multicast In Delay Tolerant Networks (MIDTONE) [Narmawala und Srivastava, 2009] gehört zu den flutenbasierten Protokollen und arbeitet nach dem Spray-and-Wait Mechanismus [Spyropoulos u. a., 2005] mit zusätzlicher Netzwerkcodierung (Definition 30). Netzwerkcodierung reduziert den Overhead und erhöht den Nachrichtendurchsatz, hat aber den Nachteil der Erhöhung der Rechenleistung sowie der Speicherkapazität in den Knoten. MIDTONE ist ein Protokoll, welches die Speicherverwaltung anhand von Löschregeln durchführt. Je nach gewählter Regel werden beispielsweise die Nachrichten zufällig gelöscht oder anhand eines ermittelten Ranges. Auch in diesem Protokoll wird auf die zuverlässige Multicastübertragung nicht eingegangen.

Definition 30 (Netzwerkcodierung) *„Im Unterschied zum klassischen Routing, bei dem Nachrichten von verschiedenen Quellen getrennt voneinander in Form von Paketen durch ein Netzwerk (wie z. B. das Internet) geschleust werden, können bei der Netzwerkcodierung die Zwischenknoten empfangene Nachrichten kombinieren, um sie dann an benachbarten Knoten weiterzuleiten. Die Hauptvorteile der Netzwerkcodierung bestehen in einem erhöhten Datendurchsatz und/oder einer Erhöhung der Zuverlässigkeit, aber auch die Datensicherheit kann verbessert werden.“ [Höher, 2013]*

Das Delay and Disruption Tolerant Multicasting Protocol (DTCAST) [Afanasyev u. a., 2009] ist ein Beispiel für intelligentes Routing. Es kombiniert selektives Routing für relative statische Netze mit flutenbasiertem Routing in hoch mobilen Netzen. Überdies bietet es zwei Klassen der zuverlässigen Datenauslieferung an: die garantierte Auslieferung und die Best Effort-Auslieferung (Definition 31) Jede Multicastnachricht ist mit einer Gruppenadresse sowie einer Liste der Zielknoten verbunden. Bei der garantierten Auslieferung erhält der vorgeschaltete Knoten Feedback in Form von Quittungen (Acknowledgments (ACKs)) und versucht somit die Zustellung der Nachrichten zu garantieren. Jeder Knoten auf dem Weiterleitungspfad speichert die Nachricht solange, bis dieser die ACKs erhalten hat, die von den Knoten gesendet wurden, die in der Liste der Zielknoten enthalten sind. Das Protokoll geht von einer Mindestgültigkeit von Nachrichten von größer einer Stunde aus. Ein Nachteil dieses Protokolls ist, dass der Sender die Empfänger im Voraus kennen muss. In relativ statischen Netzen ist die Auslieferungsrate sehr hoch und die Speicherauslastung gering. Ist das Netz hoch mobil, wird auf ein einfaches flutenbasiertes Routing umgeschaltet und somit gelten alle Vorteile bzw. Nachteile, die das Fluten mit sich bringt. Eine explizite Speicherverwaltung gibt es in diesem Fall nicht. Die Multicastnachrichten werden lediglich mit einem Zeitstempel gekennzeichnet, um veraltete Informationen zu bestimmen, welche verworfen werden können.

Definition 31 (Best Effort) *„Best Effort („größte Bemühung“) bezeichnet eine minimalistische Dienstgütezusicherung in Telekommunikationsnetzen. Der Betreiber des Netzes sagt dessen Benutzern zu, eingehende Übermittlungsanfragen schnellstmöglich und im Rahmen der ihm zur Verfügung stehenden Ressourcen nach besten Möglichkeiten zu bedienen. Best Effort ist somit eine pauschale Qualitätszusicherung, im Zusammenhang mit abgestuften Formen spricht man vom Quality of Service (QoS).“ [Wikipedia, 2014]*

Controlled Epidemic Routing for Multicast (CERM) [Abdulla und Simon, 2008] ist ein flutenbasiertes Routingprotokoll. Es nutzt zwei Zähler, Time-To-Kill (TTK) und Time-To-Live (TTL), zur Entscheidungsfindung, welche Nachricht gelöscht werden soll, wenn der Speicher voll ist. TTL ist der Hop-Zähler zwischen DTN-Knoten und TTK ist ein Zeitstempel an jeder Nachricht, der den Ablauf der Nachricht kennzeichnet. Die Synchronisation der Knoten muss bei diesem Protokoll gegeben sein. Dieses Protokoll stellt zwar einen einfachen Mechanismus zur Speicherverwaltung bereit, betrachtet

aber nicht die aktuelle Zustellrate. Durch die TTL-Begrenzung ist zwar die Speicherauslastung nicht so hoch, dies kann allerdings die Auslieferung der Nachrichten stark verzögern. Ferner wird dadurch die Zustellung an weit entfernte Knoten, die möglicherweise nur über viele Zwischenknoten zu erreichen sind, verhindert.

Context Aware Multicast Routing (CAMR) [Yang und Chuah, 2006] gehört zu den intelligenten Routingprotokollen. Dieses Protokoll erhöht seine Übertragungsleistung, wenn die Knotendichte abnimmt. Weiterhin verwaltet es Zwei-Hop-Nachbarschaftsinformationen. Es ist in der Lage, mit Hilfe von gespeicherten Koordinaten und Geschwindigkeitsinformationen von Zielknoten, die sich in dieser Zwei-Hop-Nachbarschaft befinden, die Bewegungsrichtung des Knotens, der die Nachricht trägt, so zu beeinflussen, dass dieser sich näher in Richtung dieser Zielknoten bewegt. Das Protokoll nutzt diese Knoten quasi als Fähren. Sind keine Informationen über das bzw. die Ziele vorhanden, wird explizit versucht eine Route zu finden. Diese Kombination erlaubt eine höhere Zuverlässigkeit und eine etwas niedrige Verzögerung als bei reinen selektiven Verfahren. Ein Vorteil ist auch, dass der Nachrichtenspeicher nicht übermäßig belastet wird. Da CAMR ähnlich der MANET-Protokolle arbeitet, erzielt es keine guten Ergebnisse bei sich häufig ändernder Topologie.

RelayCast [Lee u. a., 2008] verwendet ein Zwei-Hop-Schema, bei denen die Zwischenknoten verantwortlich für die Auslieferung der Multicastnachricht direkt an den Empfänger sind. Dabei verwaltet jeder Zwischenknoten eine Warteschlange für jeden Empfänger und kopiert die Multicastnachricht entsprechend. Sobald die Nachricht an ein Gruppenmitglied übermittelt wird, wird diese aus der entsprechenden Warteschlange entfernt. Durch das Kopieren der Nachricht für jede Warteschlange werden viele Ressourcen verschwendet und ein Feedback zur Auslieferung ist auch nicht vorgesehen. Dieses Protokoll besitzt keine gute Zuverlässigkeit, wenn die Zielknoten mehr als zwei Hops vom Sender entfernt sind.

Definition 32 (HELLO-Nachrichten) *HELLO-Nachrichten sind periodisch versendete Nachrichten, welche Nachbarknoten im Netz entecken sollen. HELLO-Nachrichten können beispielsweise die Adresse der Nachbarknoten sowie die Verzögerung oder auch den Overhead vom Knoten zum Nachbarknoten enthalten. [Huang und Du, 2015]*

Epidemic-based Controlled Flooding and Adaptive Multicast for Delay Tolerant Networks (ECAM) Jin u. a. [2010] kombiniert das Fluten von Multicastnachrichten mit einer hierarchische Übertragung um Speicherplatz zu sparen. Um das Fluten zu begrenzen, wird die erhaltene Multicastnachricht im Speicher abgelegt und gleichzeitig ein Zähler für diese Nachricht initialisiert. Bei jedem Weiterleiten dieser Multicastnachricht wird der Zähler inkrementiert und bei Erreichen eines Schwellwertes wird diese Nachricht gelöscht. Für die hierarchische Übertragung wird ein Feld festgelegt, welches die Anzahl der Hops, die diese Multicastnachricht von Nicht-Gruppenmitliedern übertragen werden kann, begrenzt. Dies bedeutet, wird die Nachricht durch ein Gruppenmitglied übertragen, wird dieses Feld nicht verändert. Auch hier wird die Nachricht verworfen, wenn die Anzahl der Hops einen vordefinierten Wert übersteigt. Jedoch kann eine solche Beschränkung von Hops die Zuverlässigkeit reduzieren, wenn die Anzahl der Nicht-Gruppenmitglieder nicht korrekt ausgewählt ist. Durch das frühzeitige Löschen der Nachrichten wird der Nachrichtenspeicher nur mäßig belastet. Allerdings erhöht sich die Verzögerungszeit bei der Nachrichtenübertragung, da die Wahrscheinlichkeit einen Knoten mit der entsprechenden Multicastnachricht zu treffen, geringer wird. Darüber hinaus hat ECAM keine Informationen über mögliche Gruppenmitglieder und daher keinen Lieferstatus. Allerdings bietet ECAM einen adaptiven Mechanismus. Bei hoher Knotendichte wird das periodische Senden von HELLO-Nachrichten (Definition 32) reduziert.

Encounter-Based Multicast Routing (EBMR) [Xi und Chuah, 2009] ist ein Protokoll, das aus der Kategorie der wahrscheinlichkeitsbasierten Protokolle stammt. Es erstellt einen dynamischen Multicastbaum anhand der Kontaktdaten und berechnet somit die Wahrscheinlichkeit zwischen zwei Knoten anstelle der Route mit dem kürzesten Pfad. Des Weiteren kann der Sender eine Beschränkung der Kopien der Multicastnachrichten festlegen, was zur Verkleinerung des Multicastbaumes führt. Dadurch wird die Speicherauslastung in den Knoten vermindert. Allerdings kann dieser Ansatz auch zu einer hohen Verzögerung der Nachrichtenauslieferung führen. Die Zuverlässigkeit dieses Protokolls ist nur sehr hoch, wenn die Bewegung der Knoten vorhersehbar ist.

[Lo und Luo, 2012] stellt ein Speichermanagementsystem vor. Das Quota Based Multicast Routing (QBMR) Protokoll sortiert die zu übertragenden Multicastnachrichten nach Zeitpunkt des Erhalts der Nachricht und anhand des Hop-Zählers. Aus dem daraus entstehenden Produkt wird die Reihenfolge der Übertragung festgelegt. Dabei werden die kleineren Werte zuerst übertragen. Weiterhin wird so auch die Löschreihenfolge festgelegt, wenn der Speicher voll ist. Allerdings wird die Nachricht mit dem größten

Wert zuerst gelöscht. Dieses Protokoll fordert keine explizite Gruppenverwaltung und nutzt den flutenbasierten Routingansatz. Um die Zielknoten zu bestimmen, verwaltet jeder Knoten eine Liste, die die Multicastnachrichten enthält, und für jede Nachricht die Zielknoten, die diese Nachricht schon erhalten haben. Trifft ein Knoten einen anderen Knoten, werden die Listen zusammengeführt. Ist ein Knoten Gruppenmitglied, fügt er die eigene ID der Zielliste hinzu. Die globale Netzinformation über die Gruppenmitgliedschaften, welche nach einer gewissen Zeit zu Verfügung steht, wird herangezogen, um die gelieferten Nachrichten zu löschen. Dieses Protokoll zeigt, dass eine Speichermanagementsystem bessere Ergebnisse hinsichtlich der Zuverlässigkeit erzielt, als beim reinen flutenbasierten Ansatz. Es zeigt auch, dass eine erhöhte durchschnittliche Ende-zu-Ende Verzögerung der gelieferten Multicastnachrichten im Vergleich zum puren flutenbasierten Ansatz, ohne Speichermanagement, besteht.

[Srinivasan und Ramanathan, 2010] konzentrieren sich mit ihrem Source Assured Reliability (SAR)-Ansatz auf die garantierte Nachrichtenauslieferung ohne die Erstellung von Multicastbäumen sowie ohne das Wissen über Kontaktinformation und ist somit ein flutenbasierter Ansatz. Es wird das „Generation Time Membership Modell“ verwendet, was bedeutet, dass nur Knoten die Nachrichten erhalten, die zum Zeitpunkt der Nachrichtenerstellung Gruppenmitglieder waren. Die Gruppenmitglieder, die die Multicastnachricht erhalten sollen, werden vom Sender im Nachrichtenkopf adressiert. Dies bedeutet, dem Sender müssen alle Gruppenmitglieder bekannt sein. Jeder Knoten verwaltet eine Liste, in deren alle Gruppenmitglieder, welche die Nachricht erhalten haben, aufgeführt sind. Diese Liste wird bei Kontakt mit anderen Knoten aktualisiert. Ein Knoten löscht erst dann die Nachricht, wenn laut dieser Liste alle Gruppenmitglieder die Nachricht erhalten haben oder wenn der Knoten die Nachricht an eine bestimmte Anzahl von Knoten weitergereicht hat. Problematisch ist dieser Ansatz dann, wenn nicht genügend Nachrichtenspeicher vorhanden ist. Bei diesem Protokoll wird nicht klar, was passieren soll, wenn der Speicherplatz des Knotens aufgebraucht ist. Dieser Ansatz bezieht Gruppeninformationen beim Löschen von Multicastnachrichten ein. Somit wird die Zuverlässigkeit bei der Zustellung von Multicastnachrichten erhöht. Nachteilig ist, das jede Multicastnachricht die zusätzliche Information von Gruppenmitgliedern mit sich trägt.

Ein weiteres wahrscheinlichkeitsbasiertes Protokoll, Social-Aware Multicast (SM) [Gao u. a., 2012], betrachtet Multicast in DTNs aus dem Blickwinkel von sozialen Netzwerken. Der Fokus liegt dabei auf der Ersparnis von Kosten. Dieses Protokoll stellt eine Reihe von Multicastroutingmethoden bei der Übertragung von einzelnen und mehreren

Nachrichten vor. Die Grundlage der Multicastroutingmethoden bilden zwei Verfahren, das zentralisierten Verfahren und das gemeinschaftliche Verfahren. Wird nur eine einzelne Nachricht an mehrere Empfänger weitergeleitet, findet das zentralisiere Verfahren seine Anwendung. Bei diesem Verfahren leitet der Knoten diese Nachricht anhand der Kontaktwahrscheinlichkeit (lokale Sicht) weiter. Dabei geht der Sender davon aus, dass die Gruppenmitglieder gleich verteilt sind und diese innerhalb eine bestimmten Zeit erreicht werden können. Das gemeinschaftliche Verfahren, welches verwendet wird, wenn mehrere Nachrichten von einem Sender verschickt werden, nutzt Zwischenknoten (Relais) zur Nachrichtenweiterleitung. Bei diesem Verfahren werden die Wahrscheinlichkeiten für jeden Weiterleitungsschritt bis zum Ziel betrachtet (globale Sicht). Das Ziel ist, die Anzahl der Zwischenknoten zu minimieren. Die Berechnung der Minimierung von Zwischenknoten erfordert jedoch erhöhten Rechenaufwand. Ist eine Nachricht für ein Mitglied einer anderen Gruppe gedacht, wird diese über einen zentralen Knoten (Gateway) weitergeleitet. Der Umweg von Multicastnachrichten über Gateways führt zu höheren Verzögerungszeiten. Außerdem wird durch die Kostenersparnis auch die zuverlässige Zustellung reduziert. Die Einbeziehung von Kontaktwahrscheinlichkeiten ist nur bei relativ konstanten Bewegungsmustern sinnvoll. Ein weiterer Nachteil ist der Nachrichtenstau an den zentralisierten Knoten. Zentralisierung führt auch zu Überlastung des Nachrichtenspeichers des betreffenden Knotens, wohingegen andere Knoten bessere Zustellchancen hätten.

Die meisten vorgestellten Multicastprotokolle beziehen sich fast ausschließlich auf das Routing. Der vorgestellte Ansatz von [Srinivasan und Ramanathan, 2010] zeigt, dass Speichermanagement ein wichtiger Bestandteil der zuverlässiger Nachrichtenübertragung ist. Nur wenige Protokolle stellen Algorithmen vor, wenn der Nachrichtenspeicher voll ist und somit entschieden werden muss, welche Nachricht gelöscht werden soll. [Lo und Luo, 2012] stellen einen Ansatz vor, welcher anhand des Erhalts der Nachricht sowie am Hop-Zähler eine Löschreihenfolge festlegt. Auch bei diesem Ansatz wird allerdings die tatsächliche Auslieferung der Gruppennachrichten an die Gruppenmitglieder nicht mit einbezogen. Es gibt Versuche die Zuverlässigkeit zu steigern, indem Hop-Zähler eingesetzt werden. Die Aussage, wie lang eine Nachricht unterwegs ist, hat nichts mit der Anzahl der tatsächlich gelieferten Multicastnachrichten zu tun. Hop-Zähler führen zu Situationen, bei denen Multicastnachrichten gelöscht werden, bevor weit entfernte Gruppenmitglieder die Nachricht erhalten haben. Es kann auch passieren, dass im Vorfeld nicht alle Gruppenmitglieder bekannt waren und sich der Speicherverwaltungsalgorithmus zu früh für das Löschen entscheidet. Deshalb ist die

Tabelle 3.2: Protokollvergleich

Protokoll	Klassifizierung	Speicherbedarf	Zuverlässigkeit	Verzögerung
UBR [Zhao u. a., 2005]	Selektiv	niedrig	niedrig	hoch
STBR [Zhao u. a., 2005]	Selektiv	niedrig	mittel	mittel
DTBR [Zhao u. a., 2005]	Selektiv	niedrig	mittel	mittel
GBR [Zhao u. a., 2005]	Fluten	mittel	hoch	mittel
BBR [Zhao u. a., 2005] (Multicast Epidemic)	Fluten	hoch	hoch	niedrig
OS-Multicast[Ye u. a., 2006]	Selektiv	niedrig	mittel	mittel
MIDTONE [Narmawala und Srivastava, 2009]	Fluten	mittelt	hoch	mittel
DTCAST [Afanasyev u. a., 2009]	Intelligentes	mittel	mittel	mittel
CERM [Abdulla und Simon, 2008]	Fluten	mittel	mittel	mittel
CAMR [Yang und Chuah, 2006]	Intelligentes	mittel	hoch	mittel
RelayCast [Lee u. a., 2008]	Intelligentes	mittel	mittel	hoch
ECAM [Jin u. a., 2010]	Intelligentes	mittel	mittel	hoch
QBMR[Lo und Luo, 2012]	Fluten	mittel	mittel	mittel
EBMR [Xi und Chuah, 2009]	Wahrscheinlichkeit	mittel	mittel	hoch
SM [Gao u. a., 2012]	Wahrscheinlichkeit	mittel	mittel	hoch
SAR [Srinivasan und Ramanathan, 2010]	Fluten	hoch	hoch	niedrig
RMDA [Begerow u. a., 2013]	Fluten	mittel	hoch	niedrig

Kombination einer effizienten Speicherverwaltung gepaart mit einer intelligenten Gruppenverwaltung von Vorteil, um die zuverlässige Auslieferung von Multicastnachrichten zu erhöhen. Demzufolge ist es notwendig, ein neues Protokoll zu entwickeln, bei dem den Anwendern mit Hilfe von änderbaren Parametern eine Anpassung an die jeweilige Situation ermöglicht wird.

Die Tabelle 3.2 präsentiert eine Übersicht der vorgestellten Multicastprotokolle. Jedes Protokoll wird anhand seiner Routingstrategie klassifiziert, der Speicherplatzbedarf ermittelt, die Zuverlässigkeit angegeben und die Verzögerung aufgezeigt. Jedoch legen nur das Protokoll SAR und das neu zu entwickelte Protokoll RMDA, den Focus auf die zuverlässige Auslieferung von Multicastnachrichten. Dies ist eine wichtige Eigenschaft, die ein Protokoll für die Anwendung in Katastrophenszenarien erfüllen muss. SAR ist im Falle eines vollen Nachrichtenspeichers nicht in der Lage, eine zuverlässige Auslieferung zu ermöglichen. Es ist zu erkennen, dass die flutenbasierten Ansätze die höchste Zuverlässigkeit und die niedrigste Verzögerungszeit aufweisen. Zu dieser Kategorie zählt auch das im folgenden Kapitel (Kapitel 4) vorgestellte neuartige RMDA-Protokoll. Dieses Protokoll zählt zu den flutenbasierten Protokollen mit intelligentem Speicherverwaltungsalgorithmus. Das RMDA-Protokoll ermittelt zu löschende Multicastnachrichten mit Betrachtung der zuverlässigen Zustellung.

4 Konzept

In diesem Kapitel wird das grundlegende Konzept des neuen Gruppenkommunikationsprotokoll RMDA vorgestellt. Als Erstes werden allgemeingültige Herausforderungen für die Gruppenkommunikation in MANETs mit verzögerungstolerantem Dienst dargeboten. Danach werden die Rahmenbedingungen bzw. Annahmen für das Protokoll präsentiert. Anschließend werden die einzelnen Bestandteile von RMDA detailliert dargestellt.

4.1 Allgemeine Herausforderungen

Die Aufgabe bestand darin, ein zuverlässiges Multicastprotokoll zu entwickeln, welches die Eigenschaften von MANETs und DTNs berücksichtigt. Diese Kombination der Netze stellt besondere Herausforderung an das zu entwickelnde Protokoll. In [Begerow, 2012] wurden folgende allgemeine Herausforderungen deklariert:

- „Wie reduziere ich die benötigte Bandbreite?“
- „Wie realisiere ich die zuverlässige Zustellung?“
- „Wie spare ich Energie?“ [Begerow, 2012]

Eine weitere große Herausforderung ist der Sicherheitsaspekt, der gerade in Rettungsmissionen sehr wichtig ist. Da im Zuge der Dissertation nicht alle Aspekte bearbeitet werden konnten, wurde der Fokus auf eine der größten Herausforderungen gelegt, auf die technische Realisierung der zuverlässigen Zustellung von Multicastnachrichten an alle Gruppenmitglieder ohne Betrachtung von Sicherheitsrichtlinien.

Um die zuverlässige Zustellung in DTNs realisieren zu können, sollten die folgenden Fragen beleuchtet werden:

- „Welche Gruppenadressierung verwende ich?“
- „Welche Multicastgruppen existieren?“
- „Wie kann ich Mitglied einer Gruppe werden?“ [Begerow, 2012]

Eine geeignete Gruppenverwaltung ist die Basis für das RMDA-Protokoll.

4.2 Annahmen für RMDA

Abbildung 4.1 aus [Begerow u. a., 2014a] zeigt ein schematisches Beispiel eines Katastrophenszenarios mit verschiedenen Gruppen, wie Feuerwehrmänner und Sanitäter. Es ist zu erkennen, dass eine Gruppennachricht, übertragen werden soll.

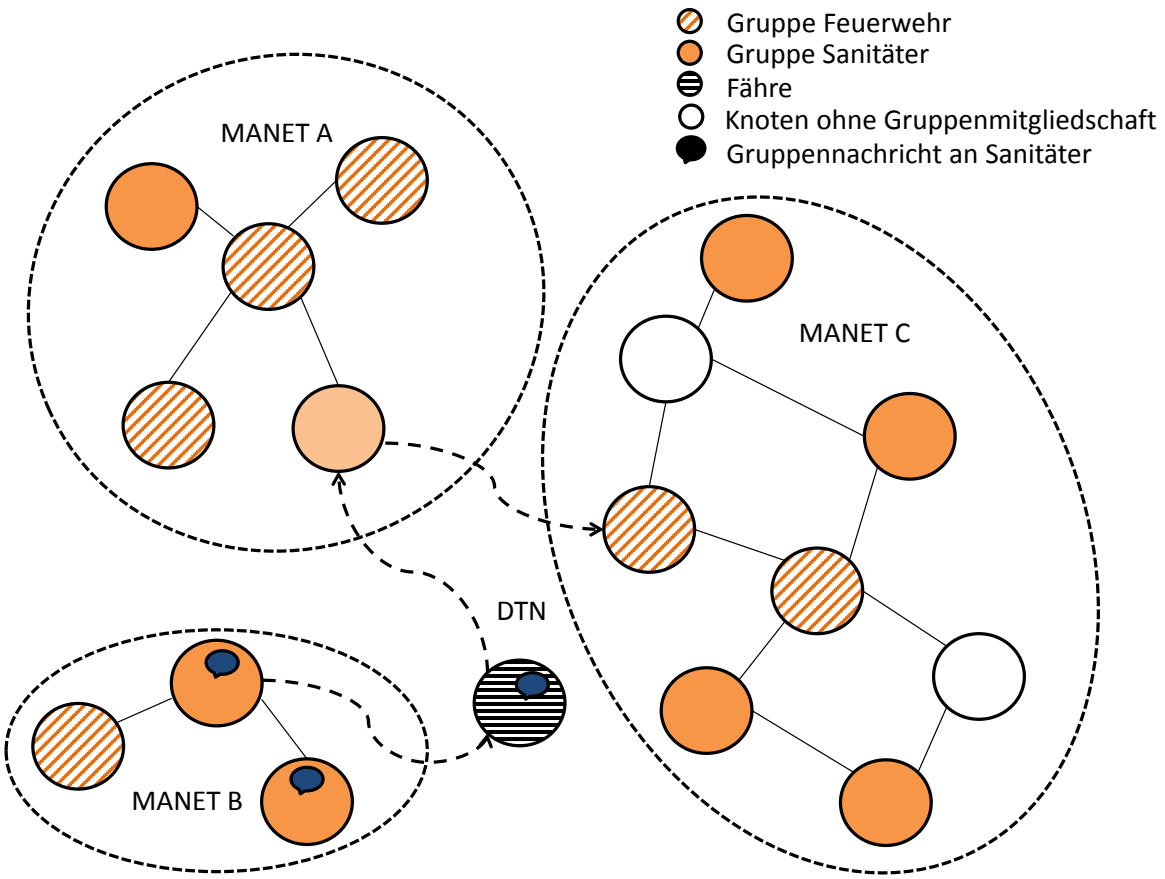

Abbildung 4.1: Katastropenszenario
[Begerow u. a., 2014a]

Mitglieder einer Gruppe, beispielsweise Feuerwehrmänner, sind interessiert an den gleichen Informationen, weshalb die Gruppenkommunikaiton einen effizienten Übertragungsmechanismus darstellt. Unicastnachrichten und Gruppennachrichten werden im Nachrichtenspeicher zwischengespeichert, bis diese ausgeliefert werden.

Das Katastrophenszenario besteht aus verschiedenen MANETs welche mittels eines verzögerungstoleranten Dienstes (DTN), dargestellt hier durch eine Fähre, verbunden sind. Diese Fähre bewegt sich zwischen den MANETs und ermöglicht so den Nachrichtenaustausch. Weiterhin besteht durch die Mobilität der Knoten die Möglichkeit, dass sich einzelne Knoten aufeinander zu bewegen und somit ein Nachrichtenaustausch realisiert wird. Der aktuelle Netzstatus sowie die aktuelle Anzahl der Gruppenmitglieder ist wegen der fehlenden durchgehenden Verbindung den einzelnen Knoten nicht bekannt, denn es können in den abgespaltenen Netzteilen Knoten einer Gruppe beitreten oder diese verlassen. Bei erneuten Kontakt zu anderen Knoten bzw. Netzteilen wird der Netzstatus und die aktuelle Gruppenmitgliedsanzahl aktualisiert. Treffen beispielsweise weitere Feuerwehrmänner am Einsatzort ein und treten der Gruppe „Feuerwehrmänner“ im MANET A bei, ist den Knoten im MANET B und MANET C dieser Gruppenbeitritt zunächst nicht bekannt. Erst nach erfolgtem Wissensaustausch, beispielsweise durch die Fähre zwischen den MANETs, ist die aktualisierte Gruppenmitgliederanzahl auch für die anderen Knoten verfügbar.

Werden Multicastnachrichten versendet, ist durch die Abspaltung von Netzteilen keine direkte Rückmeldung über die erfolgreiche Zustellung an Gruppenmitglieder möglich. Durch dieses fehlende Wissen verbleiben Multicastnachrichten „endlos“ im Nachrichtenspeicher, was zu einem Speicherplatzproblem führt.

Ist der Nachrichtenspeicher in einem Knoten voll und es sollen weiter Nachrichten empfangen werden, muss entschieden werden, welche Nachricht aus dem Nachrichtenspeicher gelöscht wird. Der Mangel an globalem Wissen erschwert dabei diese Entscheidung. Deshalb ist eine effiziente Verwaltung von Multicastnachrichten für Katastrophenszenarien notwendig, bei der eine zuverlässige Auslieferung entscheidend ist.

RMDA ist für die speziellen Anforderungen in Katastrophenszenarien entworfen worden. Die Erfahrungen zeigen, dass im Falle einer Katastrophe kaum Infrastruktur für Kommunikation, für die Bereitstellung von technischer Ausrüstung sowie für die Zustellung von Hilfsgütern besteht.

Die Gruppenanzahl ist im heutigen TETRA-Netz begrenzt und wird durch ein zentrales Gruppenkonzept bestimmt. Deshalb ist davon auszugehen, dass sich auch in Zukunft die Kommunikation in Katastrophenfällen auf nur wenige Gruppen beschränkt. Das Gruppenkonzept der BOSs ist nicht öffentlich und wird derzeit durch die autorisierten Stellen der Bundesländer und teilweise des Bundes umgesetzt. Das bedeutet, dass in dem heutigen TETRA-Netz, die Gruppen im Vorfeld festgelegt sind und zentral verwaltet werden. Zukünftig ist eine dezentrale Verwaltung mit zusätzlichen Sicherheitsfunktionen denkbar und notwendig, da durch die bestimmten Eigenschaften des DTNs, eine zentrale Verwaltung in jeglicher Hinsicht ungeeignet ist.

Folgende Annahmen werden für ein Katastrophenszenario getroffen:

- Die Infrastruktur ist teilweise oder komplett zerstört.
- MANETs ersetzen die Kommunikation.
- In den Knoten ist ein DTN bzw. RMDA integriert.
- Die Anzahl der verschiedenen Gruppen ist klein.
- Die Knoten treten einer oder mehreren Gruppen nach Ankunft an der Unglücksstätte bei.
- Ein Gruppenwechsel oder Austritt erfolgt in den seltensten Fällen während einer Mission.
- Es werden Multicastnachrichten sowie Unicastnachrichten übertragen.

Um den höchstmöglichen Anteil aller beabsichtigten Gruppenmitglieder zu erreichen und gleichzeitig den Speicher effizient zu verwalten, wurden folgende Designentscheidungen getroffen:

- Als erstes wird davon ausgegangen, dass die Knoten im DTN zunächst synchronisiert werden, beispielsweise nach [Choi und Shen, 2010]. Eine Zeitsynchronisation ist grundsätzlich auch über GPS möglich, setzt aber voraus, dass alle Knoten GPS-fähig sind und sich diese nicht in Gebäuden befinden. Die Zeitsynchronisation ist ein wichtiger Bestandteil für den in RMDA verwendeten Algorithmus, welcher Zeitstempel mit einbezieht. Allerdings ist eine enge Synchronisation im Verlauf der Mission nicht erforderlich, da Verzögerungen bei DTNs sich, in der Regel, in Größenordnungen von Sekunden oder Minuten belaufen, welche durch die Bewegungen der Knoten entstehen.

- Der Datenaustausch zwischen zwei Knoten erfolgt nach dem First in First out (FIFO)-Prinzip (Definition 33).

- Als Grundlage für den Datenaustausch dient die Multicastvariante BBR [Zhao u. a., 2005], da sie durch ihren flutenbasierten Ansatz die beste Auslieferwahrscheinlichkeit besitzt.

- BBR fordert keine globalen oder lokalen Kenntnisse über Gruppenmitgliedschaften. RMDA benötigt allerdings eine lokale Sicht der Gruppenmitgliedschaften, um den Nachrichtenspeicher optimal zu verwalten. Globales Wissen ist nicht erforderlich, wird aber annähernd erreicht, wenn sich der Mitgliederstatus der Gruppe selten ändert. Der Austausch der Gruppenmitgliedschaften erfolgt bei Kontakt mit anderen Knoten.

- Multicast- sowie Unicastnachrichten werden im selben Nachrichtenspeicher abgespeichert. Es erfolgt keine priorisierte Übertagung von Multicastnachrichten gegenüber Unicastnachrichten.

- Es wird davon ausgegangen, dass mutwilliges Modifizieren und Löschen von Nachrichten nicht erfolgt. Daher wird „bösartiges“ Verhalten von Knoten vernachlässigt. Jeder Knoten darf jeder Gruppe beitreten. Weiterhin ist jeder Knoten berechtigt, jegliche Nachricht zu erhalten und diese weiterzuleiten.

Definition 33 (FIFO) *„Kurzbezeichnung für First-in-first-out; Prioritätsprinzip (Priorität) der Warteschlangentheorie, nach dem zuerst ankommende Transaktionen zuerst bedient werden. Angewandt u.a. bei der Reihenfolgeplanung.“ [Berwanger u. a., 2015]*

Abbildung 4.2 zeigt die RMDA Architektur im Protokollstack. RMDA befindet sich innerhalb der DTN-Schicht und besteht aus zwei Modulen [Begerow u. a., 2013]. Das erste ist das Gruppenverwaltungsmodul, welches verantwortlich für die Verwaltung der Gruppen als auch für die Verwaltung der Gruppenmitglieder selbst ist. Das zweite Modul ist das Übertragungsmodul. Dieses Modul entscheidet, welche Multicastnachricht vom lokalen Nachrichtenspeicher weitergeleitet bzw. gelöscht wird. Darüber hinaus entscheidet es, ob eine eingehende Multicastnachricht für diesen Knoten bestimmt ist und an die höhere Schicht übergeben werden muss. Beide Module werden in den Unterkapiteln Gruppenverwaltungsmodul 4.3 und Übertragungsmodul 4.4 näher vorgestellt.

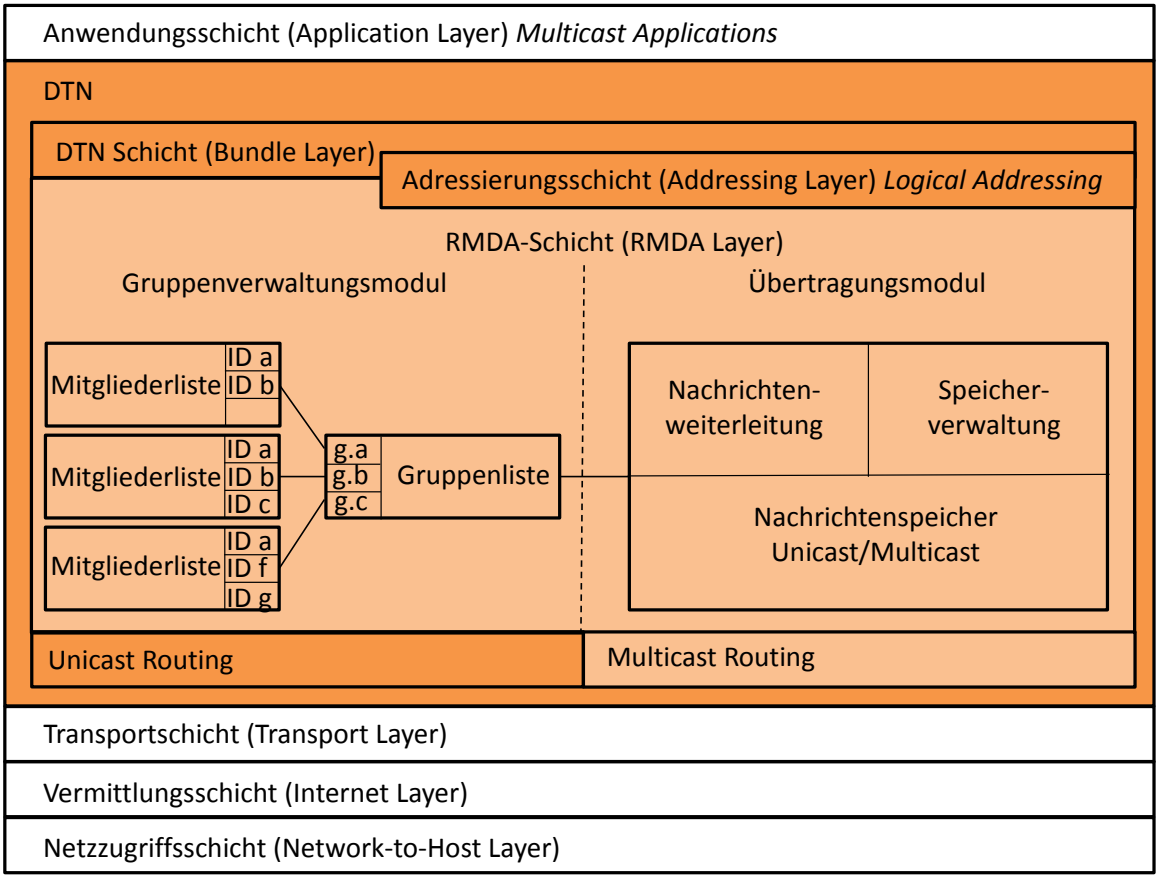

Abbildung 4.2: Internet-Referenzmodell mit RMDA

4.3 Gruppenverwaltungsmodul

Die Gruppenverwaltung bildet die Grundlage für die Speicherverwaltung des RMDA-Protokolls. Es bietet die Informationen über verfügbare Gruppen und deren Mitglieder. Ein zentraler Ansatz für die Gruppenverwaltung ist wegen der Unterbrechungen in DTNs nicht geeignet. Deshalb wird ein verteilter Ansatz verwendet. Durch die im Unterkapitel 4.2 getroffene Annahme, dass nur wenige Gruppen pro Mission existieren, wurden Listen zur Speicherung von Gruppeninformationen verwendet. Jeder Knoten speichert und verwaltet diese Managementlisten selbständig. Für die Aktualisierung dieser Managementlisten ist ein Austausch dieser im Netz erforderlich.

Das Gruppenverwaltungsmodul enthält lediglich die für die einfache technische Realisierung benötigten Funktionen, Sicherheitsmechanismen wurden nicht mit einbezogen. Jeder Knoten ist berechtigt eine Gruppe zu erstellen und kann jeder Gruppe beitreten.

4.3.1 Managementlisten

Die Managementlisten speichern Informationen über Gruppen und Gruppenmitgliedschaften. Diese Listen werden nicht im Nachrichtenspeicher des Knotens abgelegt, sondern belegen einen Speicherbereich im Knoten, der für temporäre Daten, beispielsweise für Berechnungen, vorgesehen ist. Jeder Knoten verwaltet zwei Arten von Listen, welche die lokale Sicht des Knotens auf die Gruppen und deren Mitglieder darstellen.

Die erste Liste ist die *GroupList* (Gruppenliste). Für jeden Knoten existiert nur eine Instanz dieser Liste. Diese Liste beinhaltet alle Informationen der Gruppen. Die Gruppen sind gekennzeichnet durch eine eindeutige EID, beispielsweise aus [Begerow, 2014] mit dtn://rmdagroup.feuerwehr. Folgende Felder sind Bestandteil dieser Liste:

- `GroupID` (Gruppenidentifikator)
- `CreationTime` (Erstellungszeit/Speicherzeitpunkt)
- `NodeID` (Initiatorknoten)
- `StartDate` (Startzeitpunkt)
- `EndDate` (Endzeitpunkt)

Die `GroupID` ist ein eindeutiger Gruppenidentifikator und wird im bekannten EID-Style, dem URI-Format, angegeben. `CreationTime` ist ein einmalig automatisch vergebener Zeitstempel, der das Anlegen dieser Gruppe ausschließlich in diesem Knoten kennzeichnet, d. h. jeder Knoten hat in diesem Feld den Zeitpunkt der Kenntnisnahme (Speicherzeitpunkt) dieser Gruppe. Die Liste beinhaltet zusätzlich den EID des Knotens (`NodeID`), der die Gruppe erstellt hat. `StartDate` gibt den Zeitpunkt an, zu denen Knoten der Gruppe beitreten können. Außerdem können ab diesen Zeitpunkt Multicastnachrichten an diese Gruppe versendet werden. Ein wichtiges Feld ist das `EndDate` welche den Zeitpunkt bestimmt, bis zu dem die Gruppe gültig ist, denn Gruppen werden nicht dauerhaft gelöscht, da sich Nachrichten für diese Gruppe möglicherweise noch auf dem Transport durch das Netz befinden. Ab diesem Zeitpunkt, ist es einem Knoten nicht mehr möglich einer Gruppe beizutreten. Gewöhnlich ist beim Erstellen einer Gruppe dieses Feld noch nicht belegt, da oftmals das Ende einer Mission nicht abgeschätzt werden kann, und wird deshalb meistens im Nachhinein gesetzt.

Die zweite Liste, die *MemberList* (Mitgliederliste), speichert Informationen über die Gruppenmitglieder. Dabei wird für jede Gruppe eine Liste angelegt. Diese Listen werden für die Schätzung der Anzahl der Gruppenmitglieder im Übertragungsmodul be-

nötigt. Eine *MemberList* enthält folgende Felder:

- `GroupID` (Gruppenidentifikator)
- `Member[]` (Gruppenmitglieder)
  - `NodeID` (Mitglieder EID)
  - `CreationTime` (Erstellungszeit/Speicherzeitpunkt)
  - `StartDate` (Startzeitpunkt)
  - `EndDate` (Endzeitpunkt)

Diese Liste beinhaltet wiederum den EID (`GroupID`). `Member` enthält alle Gruppenmitglieder dieser Gruppe. Für jedes Gruppenmitglied werden die EID des Knotens (`NodeID`) sowie deren Beginn der Gruppenmitgliedschaft `StartDate` vermerkt. Beim Anlegen eines neuen Datensatzes wird einmalig das Feld `CreationTime` mit dem aktuellen Zeitstempel versehen. Ist dem Knoten bekannt, wann er die Gruppenmitgliedschaft beenden möchte, setzt er das Feld `EndDate`.

Tritt ein Knoten mehrmals der gleichen Gruppe bei, bzw. aus, werden ältere Mitgliedschaften überschrieben. Dies kann dazu führen, dass eine frühere Mitgliedschaft nicht mehr nachzuvollziehen ist, und deshalb sich noch im Netz befindende Multicastnachrichten nicht zugestellt werden, obwohl beispielsweise der Knoten zum Zeitpunkt des Versendens dieser Nachricht, potentieller Empfänger war. Durch die im Unterkapitel 4.2 getroffenen Annahmen tritt ein ständiges Ein- bzw. Austreten aus einer Gruppe überhaupt nicht oder nur sehr selten auf. Deshalb und aus Speicherplatzgründen wurde auf zusätzliche Felder in der Liste verzichtet.

Wie schon erwähnt, darf in beiden Listen das `EndDate` offen gelassen werden, somit ist der Zeitpunkt der Beendigung der Gruppe bzw. der Gruppenmitgliedschaft noch nicht festgelegt.

Kommt ein Knoten mit einem Anderen in Kontakt werden die *GroupList* und *MemberLists* der Knoten aktualisiert. Dies erfolgt über den Austausch von Managementnachrichten (*MgmtMsg*).

4.3.2 Managementnachrichten

Eine Managementnachricht *MgmtMsg* dient zum Austausch der in den Knoten gespeicherten Managementlisten. Eine *MgmtMsg* wird bei neuem Kontakt zu einem Knoten

oder bei Änderung in einer der Managementlisten zu allen Nachbarknoten (Knoten in Kommunikationsreichweite) gesendet. Nach erfolgreicher Übermittlung wird diese anschließend aus dem Nachrichtenspeicher gelöscht. Der Empfangsknoten löscht die *MgmtMsg* nach Verarbeitung direkt aus dem Nachrichtenspeicher. Der Aufbau einer *MgmtMsg* wird in Abbildung 4.3 dargestellt. Die Feldgrößen aus der ursprünglichen Version [Begerow u. a., 2014a] der *MgmtMsg* wurden angepasst. Das erste Feld ist ein 2 Bit Feld und kennzeichnet, um was für einen Nachrichtentyp (Tabelle 4.1) es sich handelt. Eine *MgmtMsg* wird durch ein *00* Kennung charakterisiert. Die Nachrichtentypen Multicast und Quittung werden im Abschnitt 4.4.1 ausführlich behandelt. Die 14 Bit `MsgID` (Nachrichtenkennung) ermöglicht 16384 *MgmtMsgs*. Die Felder `SrcLength` (Senderadresslänge) und `DestLength` (Empfängeradresslänge) sind 8 Bit lang und geben an, wie lang die EID des Senders (`Source EID`) und die EID des Empfängers (`Destination EID`) ist.

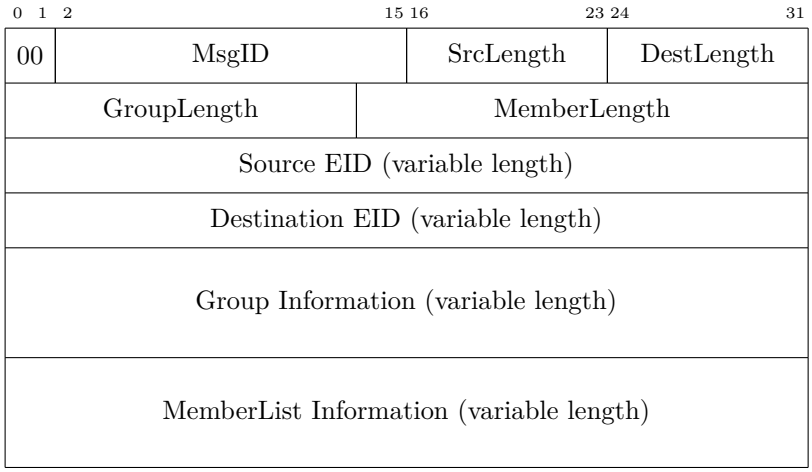

Abbildung 4.3: Struktur einer Managementnachricht

Dadurch entsteht eine Limitierung der EIDs der Knoten auf je 255 Zeichen. Eine Gesamtlänge von 16383 Zeichen des Feldes `Group Information` ermöglicht das 14 Bit Feld für die `GroupLength` (Gruppenlistenlänge). Das 18 Bit Feld der `MemberLength` (Mitgliederlistenlänge) erlaubt eine `GroupLength` (Gruppenlistenlänge) von 262143 Zeichen. Damit entsteht eine maximale Größe der *MgmtMsg* von rund 2 Mbit, was bei maximalen EIDs von ca. 255 Zeichen, 31 Gruppen in der *GroupList* und 503 Mitglieder in den *MemberLists* entspricht. Allerdings ist eine Länge der EIDs von 255 Zeichen nicht realistisch. Es ist davon auszugehen, dass eine durchschnittliche Länge von 50 Zeichen der EIDs nicht überschritten werden. Somit ergeben sich 145 Gruppen mit einer Gesamtanzahl von ca. 2340 Gruppenmitglieder.

Die *MgmtMsg* selbst enthält die *GroupList*, sowie alle *MemberLists*. Es erfolgt kein vorheriger Austausch welche Daten benötigt werden und somit kein zusätzlicher Datenverkehr.

Tabelle 4.1: Nachrichtentypen

Wert	Nachrichtentyp
00	Managementnachricht
01	Multicastnachricht
10	Acknowledgment (Quittung)

4.3.3 Funktionen

Das Gruppenverwaltungsmodul des RMDA-Protokolls stellt an der Schnittstelle zu den höheren Schichten, also zu den Anwendungen, Funktionen für das Gruppenmanagement zur Verfügung. Eine Anwendung ruft eine Funktion auf, welche das entsprechende Ereignis, beispielsweise einen Gruppenbeitritt eines Knotens auslöst. Folgende Funktionen werden an der Schnittstelle bereitgestellt:

- `CreateGroup(GroupID, StartDate, EndDate)`
- `AskGroups()`
- `ShowGroups(GroupList)`
- `JoinGroup(GroupID, Startzeitpunkt, EndDate)`
- `DeactivateGroupmebership(GroupID, EndDate)`
- `DeactivateGroup (GroupID, EndDate)`

Möchte ein Knoten respektive Anwendung eine Gruppe erstellen, ruft diese die Funktion `CreateGroup` auf. Der Knoten gibt dabei die Attribute `GroupID` (Gruppenidentifikator) sowie `StartDate` (Startzeitpunkt) und, wenn schon bekannt, das `EndDate` (Endzeitpunkt) mit. Ist das `EndDate` negativ belegt, ist die Gültigkeitsdauer dieser Gruppe nicht angegeben und kann später mit der Funktion `DeactivateGroup` mitgeteilt werden. Jeder Knoten ist berechtigt eine neue Gruppe anzulegen. Ein `CreateGroup` löst automatisch ein `JoinGroup` aus. Damit wird erreicht, dass der Ersteller der Gruppe automatisch Gruppenmitglied ist.

Die Funktion `AskGroups` ermöglicht den Knoten (bzw. den Anwendungen des Knotens) das Gruppenverwaltungsmodul nach verfügbaren Gruppen zu Fragen. Das Gruppenverwaltungsmodul antwortet mit der *GroupList* über die Funktion `ShowGroups`.

`JoinGroup` ist die Funktion zum Beitritt eines Knotens zu einer Gruppe. Dabei muss `GroupID` sowie `StartDate` mitgeben werden. Ist das `EndDate` negativ belegt, ist der Knoten solange Gruppenmitglied, bis die Gruppe selbst nicht mehr besteht.

Jederzeit ist es für einen Knoten möglich die Gruppenmitgliedschaft vorzeitig zu beenden. Dabei gibt dieser das `EndDate` und die gewünschte Gruppe (`GroupID`) der Funktion `deactivateGroupmebership` mit.

Die Funktion `DeactivateGroup`, mit ihren Attributen `GroupID` und `EndDate`, ist für das Schließen einer Gruppe zuständig. Nur der Knoten, der eine Gruppe erstellt hat, ist berechtigt, die Aktivität der Gruppe zu beenden. Dabei wird nicht die Gruppe komplett aus der Liste entfernt, sondern es wird ein `EndDate` gesetzt. Nach Setzen dieses Zeitpunkts ist der Beitritt zu dieser Gruppe nicht mehr möglich. Ist einem Knoten hingegen noch nicht bekannt, dass diese Gruppe geschlossen ist, ist diesem Knoten zunächst ein Beitritt zu dieser Gruppe möglich. Nach Listenaustausch hat jedoch das `EndDate` der *GroupList* Vorrang. Der Knoten wird nicht als Gruppenmitglied anerkannt und ist nicht berechtigt Gruppennachrichten zu empfangen. Dieser Knoten wird dann aus der *MemberList* entfernt.

4.3.4 Beispiel Gruppenverwaltung

Abbildung 4.4 zeigt ein einfaches Beispiel eines Flussdiagramms für die Gruppenverwaltung in RMDA. Dieses Beispiel aus [Begerow u. a., 2015] zeigt eine mögliche Kommunikation, die zwischen zwei Knoten, hier Knoten A und Knoten B, im Sinne der Gruppenverwaltung, stattfindet. In diesem Beispiel werden *MgmtMsgs*, die im Flussdiagramm verkürzt dargestellt werden, durch drei verschiedene Ereignisse ausgelöst:

1. Durch neuen Kontakt zwischen zwei Knoten.
2. Durch Anlegen einer neuen Gruppe.
3. Durch Beitritt eines Knotens zu einer Gruppe.

Nachdem Knoten A mit Knoten B in Kontakt gekommen ist (Abschnitt 1), sendet Knoten A eine *MgmtMsg* an Knoten B. Nach Erhalt dieser *MgmtMsg* werden die Listen miteinander verglichen (`Compare`). Werden Unterschiede in den Einträgen der Listen festgestellt, wird die betroffene Liste aktualisiert (`Update`). Ist beispielsweise nur ein Update eines Feldes eines Mitgliedes notwendig, bleibt `CreationTime` für dieses Mitglied in Knoten B unverändert. Kommt hingegen ein neues Mitglied hinzu, wird der aktuelle Zeitpunkt des Erstellens des Mitglieds in der *MemberList* in dem Feld `CreationTime` von Knoten B vermerkt. Nachfolgend generiert Knoten B eine *MgmtMsg* für Knoten A. Knoten A vergleicht und aktualisiert ebenfalls seine Listen.

Wenn eine Anwendung des Knotens eine neue Gruppe anlegt, wird die Funktion `CreateGroup` von dieser aufgerufen. In diesem Beispiel erstellt eine Anwendung des Knotens A die Gruppe *g.a* (Abschnitt 2). Hier ist das `StartDate` und `EndDate`, aus Übersichtsgründen, nicht explizit angeben. Deshalb, wird in diesem Beispiel, davon ausgegangen, dass die Gruppe *g.a* ab Zeitpunkt der Erstellung gültig ist. Innerhalb der Gruppenverwaltung wird ein neuer Eintrag in *GroupList* erzeugt (*NewEntry*). Der Ersteller der Gruppe ist gleichzeitig auch Mitglied der Gruppe und wird in die *MemberList* eingetragen (*New*). Dieses Ereignis löst wiederum eine *MgmtMsg* an den Nachbarknoten aus, in diesem Fall an Knoten B. Knoten B stellt fest (`Compare`), dass eine Gruppe hinzugekommen ist und fügt diese seiner *GroupList* hinzu (*NewEntry*). Des Weiteren erzeugt er eine neue *MemberList*, in welche die Daten des Gruppenmitgliedes kopiert werden, und versieht diese mit dem aktuellen Zeitstempel in `CreationTime` (*New*).

Im Abschnitt 3 der Abbildung 4.4 erkundigt sich eine Anwendung des Knotens B mit der Funktion `AskGroups` nach möglichen Gruppen. Die Gruppenverwaltung teilt der Anwendung, also der höheren Schicht, alle bekannten Gruppen mit. Die Anwendung des Knotens B tritt der Gruppe *g.a* mit der Funktion `JoinGroup` bei. Auch hier wird `JoinGroup` verkürzt dargestellt. Dieses Ereignis erzeugt einen neuen Eintrag in der *MemberList* mit aktuellem Zeitstempel (*New*). Durch die dadurch ausgelöste *MgmtMsg* erfährt Knoten A vom Beitritt des Knoten B zur Gruppe *g.a* (`Compare`). Knoten A erzeugt deshalb einen neuen Gruppenmitgliedseintrag (mit aktuellem Zeitstempel in `CreationTime`) in *MemberList* (*New*).

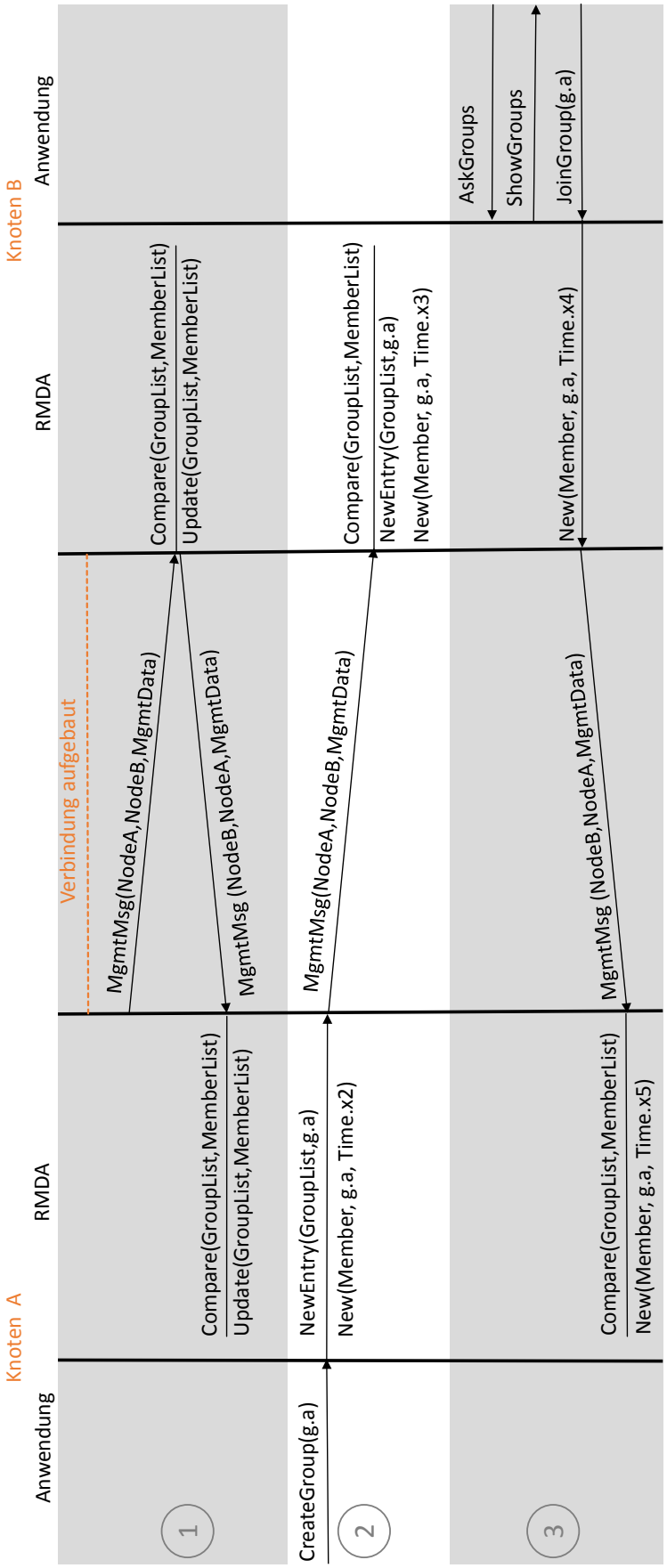

Abbildung 4.4: Weg-Zeit-Diagramm Gruppenverwaltung mit RMDA
Erweitere Version aus [Begerow u. a., 2015]

4.4 Übertragungsmodul

Das Übertragungsmodul ist verantwortlich für die Auslieferung der Multicastnachrichten an die gewünschten Gruppenmitglieder. Es entscheidet, wann welche Multicastnachricht vom lokalen Nachrichtenspeicher gelöscht werden kann und reduziert somit den benutzen Nachrichtenspeicher. Um dies zu erreichen, wird ein Algorithmus verwendet, der die aktuelle Mitgliederzahl schätzt. Diese Schätzung ist Grundlage für die Speicherverwaltungsstrategie.

4.4.1 RMDA-Nachrichten

RMDA hat außer der schon bekannten *MgmtMsg* zwei weitere Typen von Nachrichten: einmal die Multicastnachricht (*MuMsg*) selbst und die dazugehörigen Quittungen (*ACKs*).

4.4.1.1 Multicastnachricht

Wenn ein Knoten eine Nachricht an Mitglieder einer Gruppe sendet, verwendet er eine Multicastnachricht (*MuMsg*). Diese *MuMsg* ist an die Gruppenadresse (`Group EID`) adressiert und wird durch das Netz geflutet. Jeder Knoten speichert diese *MuMsg*, bis diese mittels RMDA-Algorithmus (Abschnitt 4.4.5) gelöscht wird. Abbildung 4.5 zeigt die Struktur dieser Nachricht.

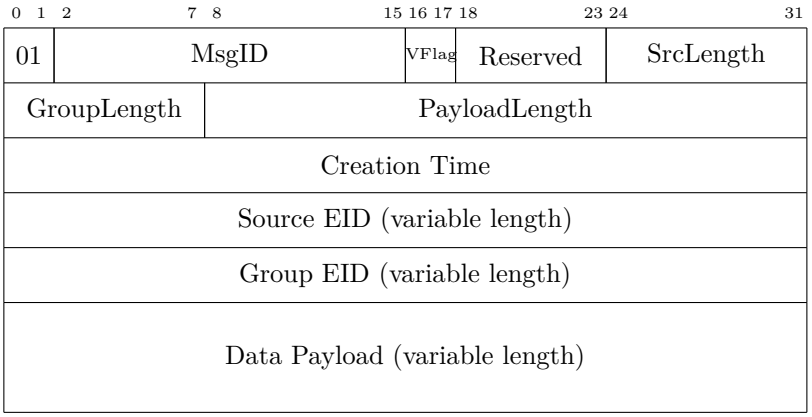

Abbildung 4.5: Struktur der RMDA-Multicastnachricht
[Begerow u. a., 2015]

Das erste Feld ist wie bei der *MgmtMsg* ein 2 Bit Feld, und kennzeichnet den Nachrichtentyp (Tabelle 4.1, Abschnitt 4.3.2). Eine *MuMsg* wird durch eine *01* Kennung charakterisiert. Die 14 Bit `MsgID` (Nachrichtenkennung) ermöglicht 16384 *MuMsgs*. Das Feld `VFlag` (Empfängerauswahl), ein 2 Bit Feld, gibt an, welche Gruppenmitglieder die Nachricht erhalten sollen. Das `VFlag` selbst wird detailliert im Abschnitt 4.4.2 erklärt. Weitere 6 Bit sind reserviert (`Reserved`). Die Felder `SrcLength` (Senderadresslänge) und `GroupLength` (Gruppenadresslänge) sind 8 Bit lang und geben an, wie lang die EID des Senders (`Source EID`) und die EID der Gruppe (`Group EID`) ist. Dadurch entsteht eine Limitierung der EIDs der Knoten auf je 255 Zeichen. Das 32 Bit Feld `Creation Time` (Erstellungszeit), gibt den Zeitpunkt der Erzeugung der *MuMsg* an. Das 24 Bit große `PayloadLength` (Nutzdatenlängenfeld) ermöglicht eine reine Nutzdatenlänge (`Data Payload`) von maximal 16777215 Zeichen. Eine *MuMsg* erreicht daher ein maximales Datenvolumen ca. 16 MB.

4.4.1.2 Quittung

Hat ein Gruppenmitglied eine *MuMsg* erhalten, sendet dieses eine Quittung (*ACK*) an die Gruppe. Jeder Knoten speichert diese *ACKs* neben den eigentlichen *MuMsgs*. Diese *ACKs* werden für den RMDA-Algorithmus (Abschnitt 4.4.5) benötigt.

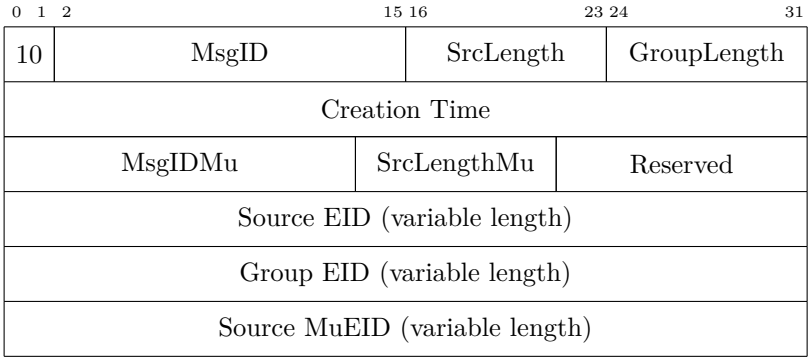

Abbildung 4.6: Stuktur der RMDA-Quittung

Eine Quittung bzw. *ACK* ist durch *10* gekennzeichnet. Die 14 Bit `MsgID` (Nachrichtenkennung) ermöglicht 16384 dieser *ACKs*. Die Felder `SrcLength` (Senderadresslänge) und `GroupLength` (Gruppenadresslänge) sind auch beim *ACK* 8 Bit lang, wodurch eine maximale Länge der Felder `Source EID` und `Group EID` auf 255 Zeichen beschränkt wird. Das 14 Bit Feld `MsgIDMu` (Nachrichtenkennung der zu quittierenden *MuMsg*) im Zusammenhang mit der `Source MuEID` (EID des Senderknotens der *MuMsg*) er-

möglichen eine eindeutige Identifizierung der zu quittierenden Multicastnachricht. Die Länge der `Source MuEID` wird durch das 8 Bit Feld `SrcLengthMu` bestimmt. Weitere 10 Bit sind reserviert (`Reserved`). Auch hier ist das 32 Bit Feld `Creation Time` (Erstellungszeit) der *ACK* angegeben. Dadurch entsteht eine maximale Quittungsgröße von rund 0,8 kB.

4.4.2 Empfängeridentifikation

Eine wichtige Rolle spielt die Empfängeridentifikation in Katastrophensituationen, welche in [Begerow u. a., 2015] vorgestellt wurde. Die Empfängeridentifikation legt diejenigen Knoten fest, die eine *MuMsg* erhalten, abhängig vom Zeitpunkt ihres Gruppenbeitritts bzw. Gruppenaustritts. Tritt beispielsweise ein Feuerwehrmann aus der Gruppe „Feuerwehr Einsatz Ilmenau“ aus, soll er trotzdem die *MuMsg* (nach Gruppenaustritt versendet) mit dem Inhalt „Gesundheitsprüfung nach Einsatz für alle beteiligten Einsatzkräfte am...“ bekommen, obwohl dieser schon lange kein Gruppenmitglied mehr ist. Ohne eine Empfängeridentifikation würden nur Gruppenmitglieder die *MuMsg* bekommen, welche zum Zeitpunkt des Sendens bzw. des Erhalts der *MuMsg* Gruppenmitglieder sind.

4.4.2.1 VFlag-Festlegungen

Das `VFlag` aus der *MuMsg* (Unterabschnitt 4.4.1.1) legt also fest, ob der Knoten ein gültiger Empfänger ist oder nicht. Die Gültigkeitsoptionen sind in der Tabelle 4.2 beschrieben.

Tabelle 4.2: VFlag Beschreibung
[Begerow u. a., 2014a]

VFlag Wert	Beschreibung
00	Gültig, wenn der Knoten beim Erhalt der Nachricht Gruppenmitglied ist.
01	Gültig, wenn der Knoten beim Erstellen der Nachricht Gruppenmitglied war.
10	Gültig, wenn der Knoten beim Erhalt sowie beim Erstellen der Nachricht Gruppenmitglied ist.
11	Immer gültig, wenn der Knoten jemals Gruppenmitglied war oder noch ist.

Das Übertragungsmodul des RMDA-Protokolls vergleicht bei Erhalt einer *MuMsg* anhand des `VFlags`, des Erstellens der *MuMsg* und anhand der Informationen aus *MemberList*, ob die *MuMsg* an die Anwendung weitergereicht wird, d. h. der Knoten ist Gruppenmitglied, oder wenn der Knoten kein Gruppenmitglied ist, ob diese nur zur Weiterleitung zwischengespeichert wird

Tritt ein Knoten nach dem Erstellen einer *MuMsg* der Gruppe bei, die diese Nachricht erhalten soll, und das `VFlag` ist auf den Wert *00* gesetzt, wird dieser Knoten als Gruppenmitglied identifiziert, obwohl dieser beim Erstellen der Multicastnachricht noch kein Gruppenmitglied war. Trifft ein Feuerwehrmann später an der Unglücksstelle ein und im Vorfeld wurde eine Gruppennachricht an die Gruppe Feuerwehr versendet, die beispielsweise Koordinaten für Sammelpunkte beinhalten, ist dies eine wichtige Information und sollte auch dem neu hinzukommenden Gruppenmitglied zugestellt werden können, wenn diese Nachricht sich noch in einem Nachrichtenspeicher eines Knotens befindet.

Eine *MuMsg* mit dem `VFlag` Wert *01* soll beispielsweise nur Gruppenmitglieder erreichen, die zum Zeitpunkt der Erstellung dieser Nachricht Mitglied der Gruppe gewesen sein müssen. Dies ist sinnvoll, wenn der Zeitpunkt des Erstellen der Nachricht wichtig für die Auswahl der Empfänger ist. Eine Nachricht, die beispielsweise um 10:05 Uhr an die Gruppe Polizisten versandt wurde, könnte lauten: „Status der bis 10:00 Uhr benötigten Reservekanister übermitteln.“ Alle später beitretenden Knoten haben bis dahin keinen Kanister benötigt und die Nachricht ist für diese uninteressant und muss deshalb nicht an die später beigetretenen Gruppenmitglieder versendet werden.

Teilweise ist es auch wichtig, dass der Knoten beim Erstellen sowie bei Erhalt der Nachricht Gruppenmitglied ist (`VFlag` Wert *10*). Werden beispielsweise Polizisten der Gruppe „Suchtrupp Polizei“ losgeschickt, um eine vermisste Person zu finden, wird nach Auffinden der Person eine Nachricht an die Gruppe gesendet, um dies mitzuteilen. Dies interessiert natürlich nur die Gruppenmitglieder, die zum Zeitpunkt des Versendens der Nachricht Gruppenmitglied waren und dies auch noch sind, wenn sie diese Nachricht erhalten.

Andererseits, in einer Situation, wie eine Kontamination ist es notwendig alle ehemaligen sowie alle aktuellen Gruppenmitglieder zu informieren. Das kann man erreichen, indem man das `VFlag` auf den Wert *11* setzt.

4.4.2.2 VFlag-Beispiele

Die Abbildungen 4.7 bis 4.10 zeigen den Einfluss des `VFlags` im Bezug auf die Auslieferung an einem einfachen Beispiel. Dieses Beispiel wurde aus [Begerow u. a., 2015] entnommen. Eine Zeitsynchronisation wird in diesem Beispiel vorausgesetzt. Die Abbildungen enthalten jeweils die selben sechs Knoten, Knoten A bis F. Alle Knoten, außer Knoten F, sind Gruppenmitglied der Gruppe *g.a*. Die Zahlenwerte stellen normalisierte Zeitstempel dar, die im realen Protokoll oder System Datums- und Zeitwerte sind. In allen 4 Fällen sendet Knoten A eine *MuMsg* zum Zeitpunkt *2* zur Gruppe *g.a* mit verschiedenen `VFlags`. Des Weiteren wird ein flutenbasierter Algorithmus, der BBR [Zhao u. a., 2005] (Multicast Epidemic), zu Grunde gelegt, bei dem eine Kopie der *MuMsg* an alle Knoten weitergeleitet wird. Diese Kopie erhält jeder Knoten zu einem anderen Zeitpunkt, Knoten F z. B. zum Zeitpunkt *4*, wie an den Pfeilspitzen angeben ist.

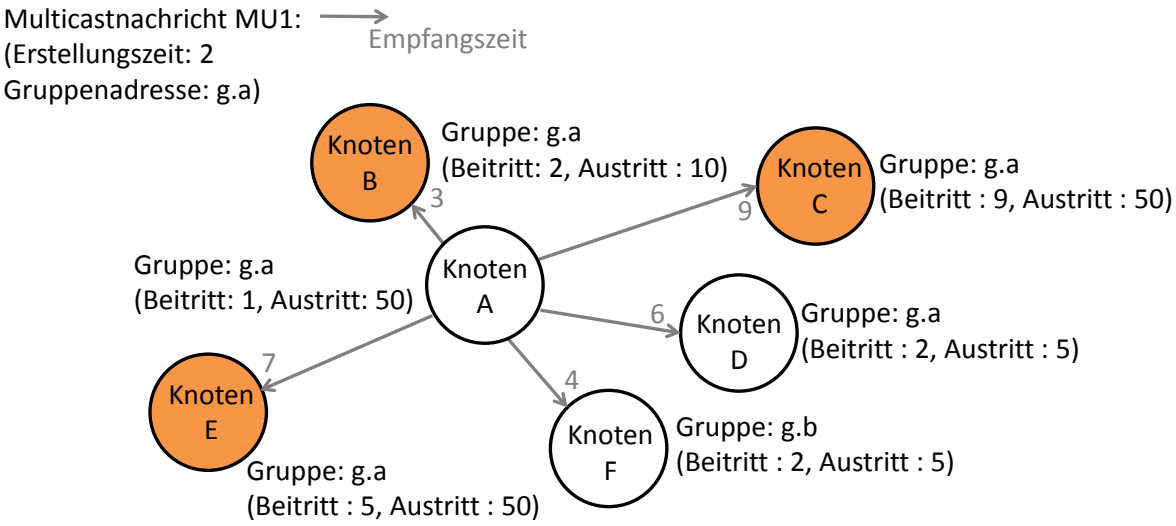

Abbildung 4.7: Multicastnachricht mit VFlag: 00
[Begerow u. a., 2015]

Die Abbildung 4.7 zeigt die Auslieferung, wenn das `VFlag` auf *00* gesetzt ist. Dies bedeutet, dass nur die Knoten die *MuMsg MU1* an eine höhere Schicht (Anwendung) weiterleiten, die beim Erhalt der *MuMsg* Gruppenmitglied sind. In diesem Beispiel sind das die Knoten B, C und E. Die anderen Knoten übermitteln *MU1* nur an andere Knoten. Knoten F ist kein Gruppenmitglied der Gruppe *g.a* und leitet deshalb die Nachricht auch nur weiter.

Nur die Knoten, die Gruppenmitglieder zum Zeitpunkt des Erstellens der *MuMsg MU2* waren, also zum Zeitpunkt *2*, erhalten die *MuMsg*, wenn das `VFlag` auf *01* gesetzt ist. Abbildung 4.8 zeigt, dass dies die Knoten B und D sind. Knoten E und C treten erst der Gruppe *g.a* bei, nachdem Knoten A die *MU2* erstellt hat.

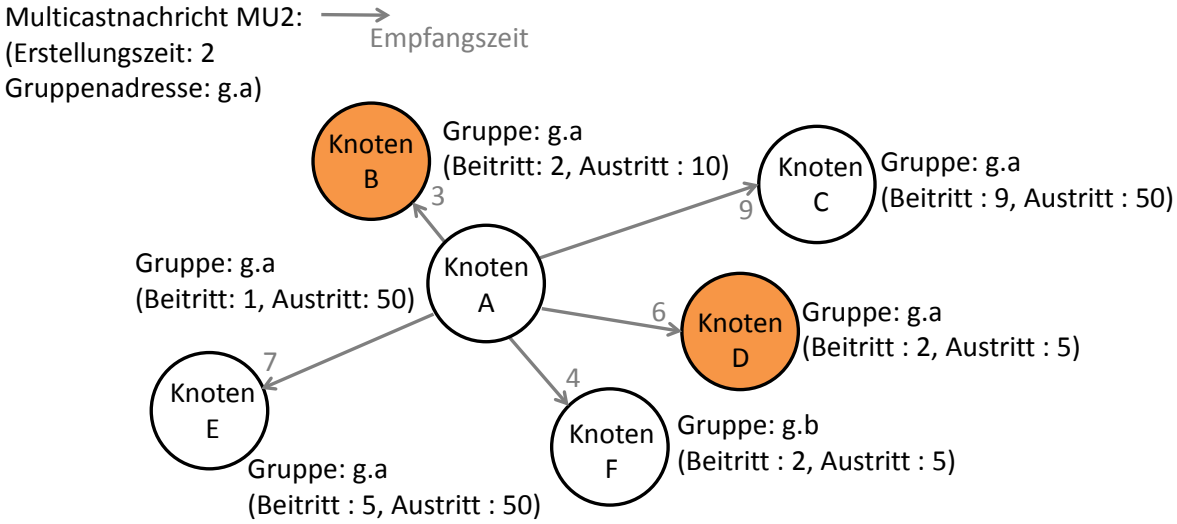

Abbildung 4.8: Multicastnachricht mit VFlag: 01
[Begerow u. a., 2015]

Abbildung 4.9 zeigt das Verhalten, wenn das `VFlag` der *MuMsg MU3* auf *10* gesetzt ist. Nur die Knoten übergeben die *MuMsg MU3* an die höhre Schicht, die gleichzeitig Gruppenmitglied zum Zeitpunkt des Erstellens, wie auch beim Erhalt von *MU3* sind. In diesem Beispiel ist dies nur Knoten B. Alle anderen Knoten sind entweder zum Zeitpunkt des Erstellens oder bei Erhalt von *MuMsg MU3* kein Gruppenmitglied der Gruppe *g.a*.

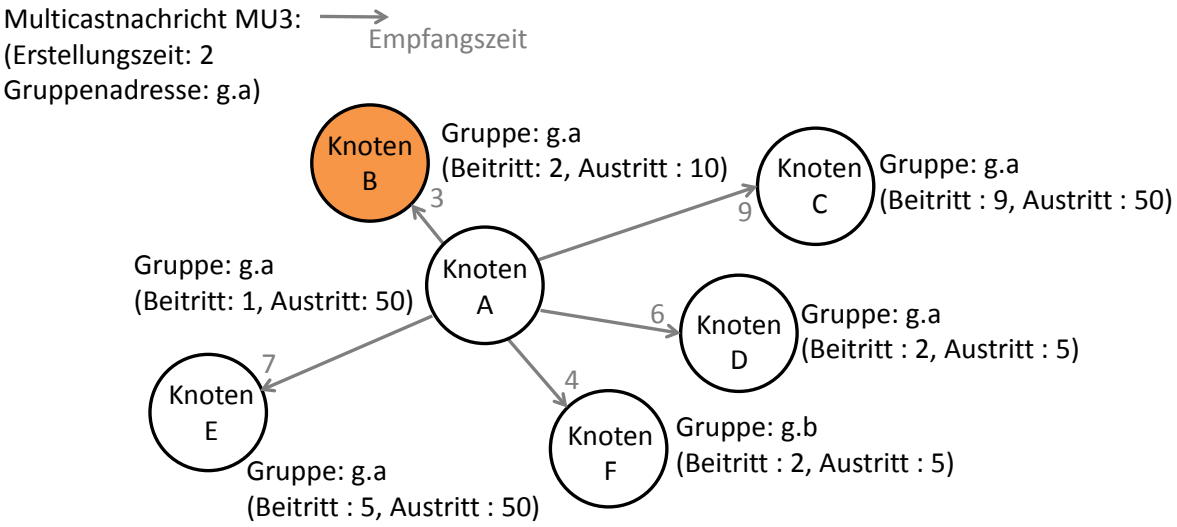

Abbildung 4.9: Multicastnachricht mit VFlag: 10
[Begerow u. a., 2015]

Ein Beispiel, wenn das `VFlag` auf *11* gesetzt ist, ist in Abbildung 4.10 zu sehen. Außer Knoten A, der Sendeknoten ist, und Knoten B, der niemals Gruppenmitglied der Gruppe *g.a* war, erhalten alle anderen Knoten die *MuMsg MU4*.

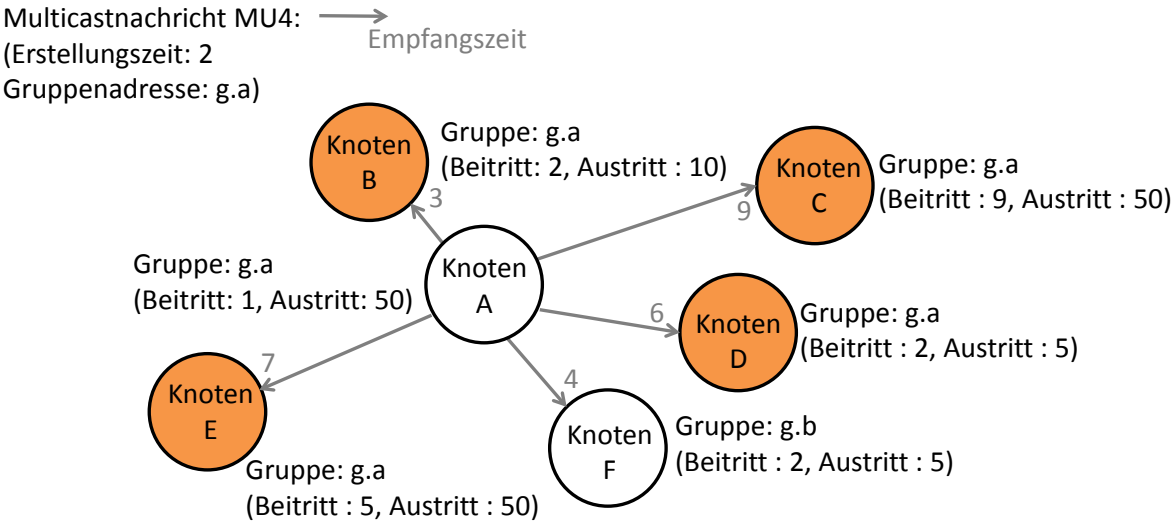

Abbildung 4.10: Multicastnachricht mit VFlag: 11
[Begerow u. a., 2015]

Diese Beispiele zeigen deutlich, den Einfluss von `VFlag` auf die Auswahl der tatsächlichen Empfänger, unabhängig von der aktuellen Anzahl der Gruppenmitglieder. Folglich hat die Empfängeridentifikation einen großen Einfluss auf den eigentlichen RMDA-Algorithmus, welcher im Abschnitt 4.4.5 ausführlich erklärt wird.

4.4.3 Speicherverwaltungsstrategie

Speicherplatz ist heutzutage kostengünstig und quasi in beliebigem Maße verfügbar. In Katastrophenszenarien ist es trotzdem notwendig Geräte mit möglichst kleinen Speichern zwischen 2 MB und 5 MB auszurüsten. Kleine Speicher ermöglichen schnelleres Abspeichern, Verarbeiten sowie Abrufen von Daten. Durch die zum Teil kurzen Kontaktzeiten zwischen den Knoten ist es nicht sinnvoll mehrere Gigabyte Nachrichten zu übermitteln. Die derzeit benutzten digitalen Handfunkgeräte als auch die digitalen Fahrzeuggeräte der BOSs werden in den nächsten Jahren nicht ausgetauscht und können lediglich neu konfiguriert werden. Sie bieten deshalb nur geringe Speichermöglichkeiten. Deshalb ist eine effiziente Speicherverwaltung notwendig.

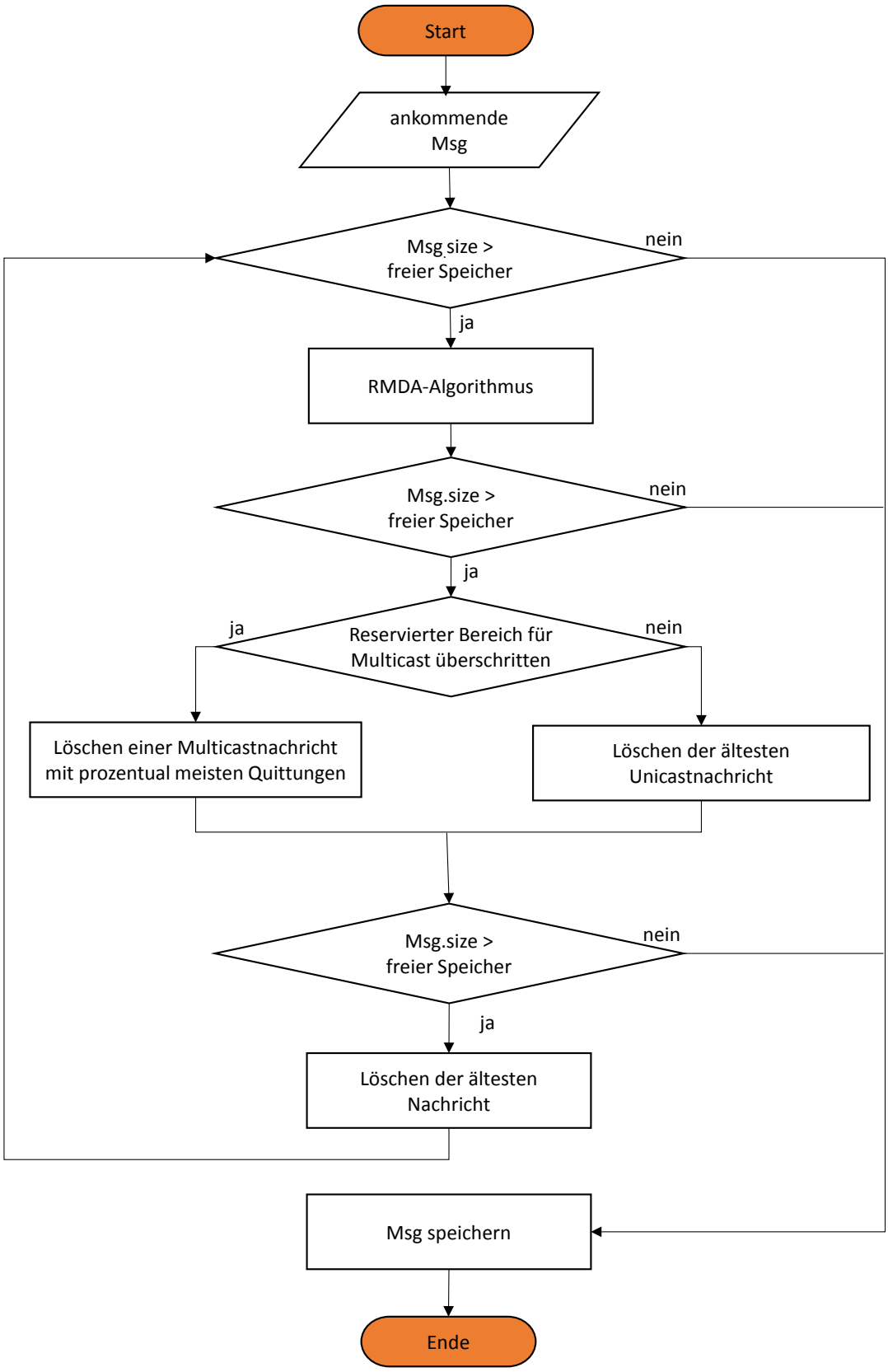

Abbildung 4.11: Speicherverwaltungsstrategie mit RMDA

Die Speicherverwaltung erfolgt unabhängig von anderen Knoten. Die Abbildung 4.11 zeigt den schematischen Ablauf. Es wird angenommen, das *MuMsgs* sowie Unicastnachrichten (*UMsgs*) gemeinsam in einem Speicher abgelegt werden. Dabei ist je nach ausgewähltem Level (Details sind im Abschnitt 4.4.4 beschrieben) prozentual Speicherplatz für den jeweiligen Nachrichtentyp reserviert. Wichtig ist ebenfalls die Speicherung der Nachrichten vom Typ *ACK*, da diese ebenfalls an andere Knoten weitergeleitet werden müssen.

Wenn ein Knoten eine Nachricht *Msg* empfängt, dabei spielt es keine Rolle, ob dies eine *MuMsg*, ein *ACK* oder eine *UMsg* ist, wird überprüft, ob für diese *Msg* ausreichend Speicherplatz vorhanden ist. Ist das der Fall, kann die *Msg* direkt gespeichert werden. Wenn nicht, müssen aus dem Nachrichtenspeicher solange Nachrichten gelöscht werden, bis ausreichend Speicherplatz für *Msg* vorhanden ist.

Als erstes wird der RMDA-Algorithmus (siehe Abschnitt 4.4.5) ausgeführt. Dies ist der Hauptansatz des RMDA-Protokolls, der gezielt die *MuMsgs* und die dazugehörigen *ACKs* ermittelt bzw. löscht, bei denen die gewünschte Auslieferungsrate laut eingestelltem Level erreicht wurde. Ist eine Löschung erfolgt, wird geprüft, ob der freigewordenen Speicher jetzt ausreicht und somit die *Msg* gespeichert werden kann.

Ist durch den RMDA-Algorithmus keine Löschung erfolgt, wird anhand der prozentual zugeordneten Speicherplatzauslastung (Abschnitt 4.4.4) entschieden, ob eine *MuMsg* oder eine *UMsg* gelöscht werden soll. Ist der Speicherplatzanteil der *MuMsgs* ausgeschöpft, wird der Nachrichtenspeicher nach der *MuMsg* durchsucht, die den höchsten Prozentsatz *ACKs* erhalten hat. Diese *MuMsg* wird mit ihren zugehörigen *ACKs* entfernt. Ist der hingegen der Speicherplatzanteil *UMsgs* erreicht bzw. überschritten, wird die älteste *UMsgs* gelöscht.

Konnte auch in diesem Fall keine *Msg* gelöscht werden, wird die älteste Nachricht, die sich im Speicher befindet entfernt. Dieser Fall kann auftreten, wenn sich beispielsweise nur *ACKs* im Nachrichtenspeicher befinden.

Wenn durch die oben beschriebene Prozedur nicht genug Speicherplatz für *Msg* entsteht, wird sie solange wiederholt, bis ausreichend Speicherplatz für *Msg* vorhanden ist.

4.4.4 RMDA-Speicherplatzanteilermittlung

Wie schon erwähnt, werden alle Nachrichtentypen in einen gemeinsamen Nachrichtenspeicher abgelegt. Das RMDA-Protokoll bietet die Möglichkeit, *MuMsgs* auf verschiedenen *Levels* bevorzugt zu behandeln. Dadurch wird den Anwendern erlaubt, das Protokoll an die jeweilige Situation anzupassen. Dabei werden für die Nachrichtentypen *MuMsgs* und *UMsgs* Speicherplatz reserviert. *ACKs* werden zum Speicherplatzanteil der *MuMsgs* hinzugerechnet. Die Speicherplatzeinschränkungen in jedem Level gelten nur, wenn *MuMsgs* und *UMsgs* vorhanden sind und der Nachrichtenspeicher aufgebraucht ist. Solange genügend Speicherplatz verfügbar ist, kann jeder Nachrichtentyp mehr Platz als das definierte Verhältnis belegen. So wird verhindert, dass Ressourcen verschwendet werden. Tabelle 4.3 (erstmals vorgestellt in [Begerow u. a., 2015]) stellt drei verschiedene *Levels* vor.

Tabelle 4.3: RMDA Verteilung
[Begerow u. a., 2015]

Level	Erhaltene ACKs	Anteil Multicast	Anteil Unicast
1	100 %	90 %	10 %
2	75 %	80 %	20 %
3	50 %	70 %	30 %

Level 1 ist so ausgelegt, dass man eine sehr hohe Zustellrate von Multicastnachrichten erzielen kann. Die optimale Kombination von Speicherplatzanteil und Anteil der zu erwartenden ACKs wurde experimentell ermittelt (siehe dazu Abschnitt 5.3.3). Mit *Level 1* werden erst dann *MuMsgs* gelöscht, wenn die durch Schätzung ermittelte erwartete Anzahl *ACKs* (laut RMDA-Algorithmus des Abschnitts 4.4.5) bei 100 % liegen. Sollten wie in Abbildung 4.11 durch den regulären RMDA-Algorithmus keine Ergebnisse erzielt worden sein, werden erst *MuMsgs* gelöscht, wenn diese 90 % des Gesamtanteils aller Nachrichten erreicht haben und es erforderlich ist, Nachrichten aus dem Nachrichtenspeicher zu löschen. Tritt dieser Fall ein, wird eine Multicastnachricht mit ihren dazugehörigen *ACKs* gelöscht, welche noch keine 100 % *ACKs* erhalten hat. In *Level 1* liegt der Anteil der gespeicherten *UMsgs* bei nur 10 % bei maximaler Ausnutzung des Speicherplatzanteils von *MuMsgs*. Da eine besonders hohe Zuverlässigkeit von *MuMsgs* im Katastrophenfall erwartet wird, ist dies die empfohlene Einstellung.

Level 2 und *Level 3* verringern deshalb den „quasi“ reservierten Nachrichtenspeicher für *MuMsgs* auf jeweils 80 % und 70 % und erlauben somit einen höheren Anteil von *UMsgs*. Auch hier werden nur Nachrichten aus dem Speicher entfernt, wenn der Nachrichtenspeicher voll ist. Befinden sich beispielsweise nur 1 % *MuMsgs* im Nachrichtenspeicher, ist es möglich, dass die restlichen 99 % durch *UMsgs* belegt werden können. Allerdings werden dann, im Vergleichsfall, solange *UMsgs* gelöscht, bis beispielsweise deren Anteil auf 20 % bei *Level 2*-Einstellung gesunken ist.

Tabelle 4.3 zeigt auch, dass die jeweils erwarteten *ACKs* verringert werden. Dies ermöglicht ein früheres Löschen der *MuMsgs* beispielsweise schon bei 50 % der erhalten *ACKs*. Hier wird der Effekt zu Nutze gemacht, dass andere Knoten die Nachricht an potenzielle Empfänger weiter leiten können.

4.4.5 RMDA-Algorithmus

Der RMDA-Algorithmus bildet den Kern für die Entscheidung, wann welche Nachricht gelöscht wird. Dieser Algorithmus schätzt die Zustellrate anhand der ermittelten Mitglieder einer Gruppe und der zu einer *MuMsg* erhaltenen *ACKs* ($count\{ack\}$). Zum besseren Verständnis wurde der Algorithmus mit einem kleine Beispiel untermauert.

Wie schon erwähnt, ist die Empfängeranzahl der Gruppenmitglieder abhängig vom `VFlag` der *MuMsg* (msg) bzw. der Sendezeit, siehe dazu Tabelle 4.2 aus dem Abschnitt 4.4.2. Die Gruppenzugehörigkeit eines Knotens $IsMember$ (Formel 4.1) wird ermittelt anhand des Zeitstempels des Erstellens von *MuMsg* ($msg.t_{create}$) sowie des Zeitpunktes des Erhalts der *MuMsg* ($msg.t_{receive}$). $Mlist[m]$ ist der zu prüfende Gruppenmitgliedseintrag aus der aktuellen *MemberList* ($mlist$). Der Index m ist der laufende Index von $mlist$. Die Funktion *IsMember* gibt *true* für „ist Gruppenmitglied“ und *false* für „ist kein Gruppenmitglied“ zurück.

Die Tabelle 4.4 zeigt ein Beispiel einer *MemberList* ($mlist$). Sie enthält die `NodeID` (Mitglieder EID), die `CreationTime` (Erstellungszeit/Speicherzeitpunkt), `StartDate` (Startzeitpunkt der Gruppenmitgliedschaft), das `EndDate` (Endzeitpunkt der Gruppenmitgliedschaft) sowie zur besseren Übersicht beispielhaft einen Index. Es wird angenommen, dass dieses Beispiel einer Gruppe zugeordnet ist.

$$IsMember(mlist[m], msg)$$

$$IsMember = \begin{cases} (mlist[m].join \leq msg.t_{received}) & \\ \quad \wedge [(mlist[m].leave > msg.t_{received}) & \\ \quad\quad \vee (mlist[m].leave = -1)] & \text{if} \quad VFlag = 00 \\ (mlist[m].join \leq msg.t_{create}) & \\ \quad \wedge [(mlist[m].leave > msg.t_{create}) & \\ \quad\quad \vee (mlist[m].leave = -1)] & \text{if} \quad VFlag = 01 \\ \{(mlist[m].join \leq msg.t_{received}) & \\ \quad \wedge [(mlist[m].leave > msg.t_{received}) & \\ \quad\quad \vee (mlist[m].leave = -1)]\} & \\ \quad \wedge \{(mlist[m].join \leq msg.t_{create}) & \\ \quad \wedge [(mlist[m].leave > msg.t_{create}) & \\ \quad\quad \vee (mlist[m].leave = -1)]\} & \text{if} \quad VFlag = 10 \\ true & \text{else} \end{cases} \tag{4.1}$$

Ist beispielsweise das VFlag von *MuMsg* mit $msg.t_{create}$ drei auf den Wert *01* gesetzt, würde $IsMember$, laut Beispiel aus Tabelle 4.4, bei den Knoten A, Knoten B, Knoten C und Knoten F, *true* zurückgeben, wohingegen die anderen Knoten den Rückgabewert *false* liefern.

Der Zeitstempel k_{create} (Formel 4.2) gibt den Zeitpunkt des Listenaustauschs an, der den minimalen Abstand zu $msg.t_{create}$ hat. Beziehungsweise k_{create} ist die `CreationTime` (Erstellungszeit/Speicherzeitpunkt) aus *MemberList* eines Mitglieds oder (bei gleicher `CreationTime`) mehrere Mitglieder, die der Nachrichtenerstellung ($msg.t_{create}$) am nächsten ist. Anhand dieses Zeitstempels kann die Anzahl von Gruppenmitgliedern identifiziert werden, die dem Knoten zu diesem Zeitpunkt bekannt waren. Ist der Abstand von zwei aufeinander folgenden Zeitpunkten (`CreationTime`) zu $msg.t_{create}$ gleich, wird der aktuellere Zeitpunkt ausgewählt. Laut Beispiel aus Tabelle 4.4 wäre es die `CreationTime` sechs bei $msg.t_{create}$ von fünf.

Tabelle 4.4: Beispiel einer Mitgliederliste einer Gruppe

Index	`NodeID` (Mitglieder EID)	`CreationTime` (Erstellungszeit/Speicherze	`StartDate` (Startzeitpunkt)	`EndDate` (Endzeitpunkt)
1	Knoten A	2	2	-1
2	Knoten B	4	3	-1
3	Knoten C	4	3	-1
4	Knoten D	4	4	-1
5	Knoten E	6	6	-1
6	Knoten F	6	2	-1
7	Knoten G	7	4	-1
8	Knoten H	8	5	-1
9	Knoten I	8	4	-1
10	Knoten J	8	5	-1
11	Knoten K	8	6	-1
12	Knoten L	8	4	-1

Vergleicht man die Gruppenmitgliedsanzahl vom `CreationTime` 6 (aktuellere Zeitpunkt) zu `CreationTime` 4 (Zeitpunkt davor), ist diese unterschiedlich. Dadurch kann der zu berechnende Trend beeinflusst werden. Erfolgte eine große Änderung bei der Gruppenmitgliedsanzahl zwischen diesen zwei Zeitpunkten, wird dies bei der Ermittlung des Trends nicht berücksichtigt. Jedoch entsprechen Informationen bzw. die aktuelle Gruppenmitgliedszahl, die aus dem aktuellen Zeitstempel gewonnen werden, eher der aktuellen Situation.

$$\begin{aligned} k_{create} = mlist[i].create \quad | \quad & \forall j, 0 \leq j \leq mlist.size, j \neq i \\ & max_i(|mlist[i].create - msg.t_{create}| \\ & \leq |mlist[j].create - msg.t_{create}|) \end{aligned} \tag{4.2}$$

$$meanMember = \frac{\sum \text{Algorithmus(Summe Mitglieder zum Zeitpunkt [t])}}{\text{Algorithmus(Anzahl Zeitstempel)}} \tag{4.3}$$

Die Berechnungsschritte (Formel 4.3 bis Formel 4.4) werden für die Ermittlung des Trends (*trend*) in der Formel 4.5 benötigt. Hierfür wird die mittlere Anzahl der Gruppenmitglieder (*meanMember*) ermittelt (Formel 4.3). Dabei wird ein Gruppenmitglied immer anhand des VFlags identifiziert (Formel 4.1). Diese Abfrage wird aus Übersichtsgründen in den folgenden Formeln bzw. Algorithmen nicht explizit aufgeführt, wird aber jedes mal angewendet.

Algorithmus 1 Anzahl Zeitstempel

```
count = 1                                          ▷ Mindestens einer ist in Liste
for i = 1; i ≤ mlist.size do
    if mlist[i].create ≠ mlist[i − 1].create then
        count ++
    end if
end for
```

Algorithmus 2 Summe Mitglieder zum Zeitpunkt [t]

```
total = 0                                                    ▷ Gesamtsumme
currentMember = 1
for i = 1; i ≤ mlist.size do
    if mlist[i].create == mlist[i − 1].create then
        currentMember ++
    else
        total += currentMembers++                  ▷ plus aktuelles Mitglied
    end if
end for
total += currentMembers++                                  ▷ plus letzte Liste
```

Die Tabelle 4.5 zeigt die Aufschlüsselung der Mitgliederanzahl zu den unterschiedlichen Zeitpunkten (`CreationTime`). Diese Tabelle wurde anhand des Beispiels, bezogen auf Tabelle 4.4, unter Annahme, dass das `VFlag` der *MuMsg* auf 11 gesetzt ist, erstellt. Somit ergibt sich *meanMember* (Formel 4.3) von 6 bei einer *Summe Mitglieder Zeitpunkt [t]* (Algorithmus 2) von 30 und *Anzahl Zeitstempel* (Algorithmus 1) von fünf.

Tabelle 4.5: Mitgliederanzahl zu verschiedenen Zeitpunkten

Zeitpunkt 2	Zeitpunkt 4	Zeitpunkt 6	Zeitpunkt 7	Zeitpunkt 8
Knoten A	Knoten A	Knoten A	Knoten A	Knoten A
	Knoten B	Knoten B	Knoten B	Knoten B
	Knoten C	Knoten C	Knoten C	Knoten C
	Knoten D	Knoten D	Knoten D	Knoten D
		Knoten E	Knoten E	Knoten E
		Knoten F	Knoten F	Knoten F
			Knoten G	Knoten G
				Knoten H
				Knoten I
				Knoten J
				Knoten K
				Knoten L

Des Weiteren wird die mittlere Anzahl der Gruppenmitglieder ab k_{create} benötigt ($meanMemberK$). Dabei werden auch hier alle Gruppenmitglieder mittels des VFlags identifiziert (Formel 4.1).

$$meanMemberK = \frac{\sum \text{Algorithmus(Summe Mitglieder zum Zeitpunkt [t] } \geq k_{create})}{\text{Algorithmus(Anzahl Zeitstempel} \geq k_{create})} \tag{4.4}$$

Algorithmus 3 Summe Mitglieder zum Zeitpunkt [t] >= k_{create}

```
total = 0                                                    ▷ Gesamtsumme
currentMember = 1
for i = 1; i ≤ mlist.size do
    if mlist[i].create == mlist[i − 1].create then
        currentMember ++
    else
        if mlist[i].create > k_create then
            total += currentMembers
        end if
        currentMembers++                                     ▷ plus aktuelles Mitglied
    end if
end for
total += currentMembers++                                    ▷ plus letzte Liste
```

Algorithmus 4 Anzahl Zeitstempel $>= k_{create}$

```
count = 1                                          ▷ Mindestens einer ist in Liste
for i = 1; i ≤ mlist.size do
    if (mlist[i].create ≠ mlist[i − 1].create) ∧ (mlist[i].create > k_create) then
        count ++
    end if
end for
```

Die Tabelle 4.6 zeigt die Aufschlüsselung der Mitgliederanzahl ab Zeitpunkt k_{create}, hier als Beispiel Zeitpunkt sechs. Auch diese Tabelle wurde anhand des Beispiels, bezogen auf Tabelle 4.4, unter Annahme, dass das `VFlag` der *MuMsg* auf 11 gesetzt ist, erstellt. Somit ergibt sich $meanMemberK$ (Formel 4.4) von 8,3 bei einer *Summe Mitglieder Zeitpunkt [t] >= k_{create}* (Algorithmus 3) von 25 und der *Anzahl Zeitstempel >= k_{create}* (Algorithmus 4) von drei.

Tabelle 4.6: Mitgliederanzahl ab Zeitpunkt 6

Zeitpunkt 6	Zeitpunkt 7	Zeitpunkt 8
Knoten A	Knoten A	Knoten A
Knoten B	Knoten B	Knoten B
Knoten C	Knoten C	Knoten C
Knoten D	Knoten D	Knoten D
Knoten E	Knoten E	Knoten E
Knoten F	Knoten F	Knoten F
	Knoten G	Knoten G
		Knoten H
		Knoten I
		Knoten J
		Knoten K
		Knoten L

$$trend = meanMember - meanMemberK \tag{4.5}$$

Der eigentliche Trend wird mit Hilfe der Formel 4.5 ermittelt. Die Ermittlung erfolgt anhand der durchschnittlichen Gruppenmitgliederzahl über den gesamten Zeitraum ($meanMember$) der Gruppe, abzüglich der durchschnittlichen Mitgliederzahl der ab k_{create} ermittelten Mitglieder ($meanMemberK$). Es ergibt sich ein negativer Wert, wenn Knoten zu einer Gruppe beitreten, nachdem eine Multicastnachricht erstellt wurde. Ein positiver Trend entsteht, wenn Knoten die Gruppe verlassen. Das bedeutet, je

mehr Knoten die Gruppe verlassen, desto größer wird der Bereich (*range*). Aus dem Beispiel ergibt sich daher ein Trend von - 2,3.

Der Korrekturwert *corr* (Formel 4.6) wird ermittelt anhand der durchschnittlichen Anzahl von Gruppenmitgliedern (*meanMember*) und der gewünschten Zustellrate (*ratio*). Diese Zustellrate wird anhand des eingestellten Levels ausgewählt. Ist das Level 2 eingestellt, ist *ratio* 75 Prozent. Siehe dazu Tabelle 4.3, Spalte „Erhaltene ACKs“ des Abschnitts 4.4.4. Aus dem Beispiel ergibt sich ein *corr* von 1,5 bei *meanMember* von sechs und *ratio* von 75 Prozent.

$$corr = meanMember * \frac{100 - ratio}{100} \tag{4.6}$$

Basierend auf diesen Informationen, ermittelt der Knoten einen Bereich (*range*) wie in Formel 4.7 dargestellt. Dieser Bereich gibt die maximale Anzahl der *ACKs* an, die fehlen dürfen, damit die Nachricht gelöscht wird. Ist der Trend (*trend*) positiv, d. h. Knoten haben die Gruppe nach dem Senden der Nachricht verlassen, wird die Anzahl der ausgetreten Knoten bzw. Gruppenmitglieder auf den Korrekturwert (*corr*) addiert und somit der Bereich (*range*) erhöht. Da im Beispiel der *trend* (-2,3) negativ ist, ergibt sich ein *range* von 1,5. Dieser *range* wird aufgerundet (siehe Formel 4.8).

$$range = \begin{cases} trend + corr & \text{if} \quad trend > 0 \\ corr & \text{else} \end{cases} \tag{4.7}$$

Für die weitere Berechnung wird die aktuelle Mitgliederanzahl des Zeitpunktes benötigt, welcher kleiner oder gleich k_{create} ist, d. h. alle Mitglieder, die (abhängig vom `VFlag`) zu diesem Zeitpunkt bekannt sind. Siehe dazu Algorithmus 5. Laut Beispiel enthält k_{create} den Wert sechs. In Tabelle 4.4 ist dies die dritte Spalte, welche sechs Mitglieder beinhaltet.

Algorithmus 5 Mitgliederanzahl zum Sendezeitpunkt

```
count = 0
i = 0
while mlist[i++].create; i ≤ k_create do
    count ++
end while
```

Am Ende werden die erhaltenen *ACKs* mit der Anzahl im Algorithmus 5 ermittelten Gruppenmitglieder minus des ermittelten *range* verglichen. Ist die Formel 4.8 wahr, wird angenommen, das die *MuMsg* erfolgreich ausgeliefert wurde. Für diese Entscheidung wurde die Anzahl der gewünschten Empfänger, abhängig vom gesetzten `VFlags` als auch die gewünschten Zustellrate *ratio* mit einbezogen. Nun kann diese *MuMsg* mit den dazugehörigen *ACKs* gelöscht werden. In dem vorgestellten Beispiel wird also die *MuMsg* gelöscht, wenn der Knoten drei *ACKs* erhalten hat.

$$count\{acks\} \geq \text{Algorithmus(Mitgliederanzahl zum Sendezeitpunkt)} - \lceil range \rceil - 1 \tag{4.8}$$

4.4.6 Beispiel Übertragungsmodul

Abbildung 4.12 ([Begerow u. a., 2015]) zeigt den Nachrichtenaustausch zwischen zwei Knoten (Knoten A und Knoten B). Nachdem diese beiden Knoten ihre *MgmtMsgs* (siehe Abbildunge 4.4) ausgetauscht haben, erfolgt die eigentliche Nachrichtenübertragung.

In diesem Beispiel erstellt die Anwendung des Knoten B die *MuMsg MU1* (`SendMuMsg`) an die Gruppe *g.a*. Das Übertragungsmodul speichert die Nachrichten im Nachrichtenspeicher (`Store`). Dafür wird zuerst geprüft, ob genügend freier Speicher zur Verfügung steht und wenn nötig Nachrichten gelöscht (`Algorithm`). Knoten A stellt ebenfalls Speicherplatz zur Abspeicherung von *MuMsg MU1* zur Verfügung (`Algorithm/Store`). Nach dem die Nachricht übertragen und gespeichert wurde, überprüft das Übertragungsmodul, ob Knoten A Gruppenmitglied dieser Gruppe ist (`GroupMember`). Dies erfolgt über eine Schnittstelle zum Gruppenverwaltungsmodul, wo unter Beachtung des *VFlags* von *MuMsg MU1* und der *MemberList*, der Status *true* für „ist Gruppenmitglied“ oder *false* für „ist kein Gruppenmitglied“ zurückgeben wird. In diesem Beispiel ist Knoten A Gruppenmitglied. *MuMsg MU1* wird an die Anwendung weitergereicht (`ReceiveMuMsg`). Die Anwendung bestätigt diese mit einem *ACK* an die Gruppe *g.a* (`SendACK`). Erhält ein Knoten eine Nachricht von einer höheren Schicht (Anwendung) oder einem anderen Knoten, dabei spielt es keine Rolle, ob es eine *UMsgs*, *MuMsg* oder ein *ACKs* ist, wird nach der Speicherverwaltungsstrategie von RMDA (Abbildung 4.11) vorgegangen.

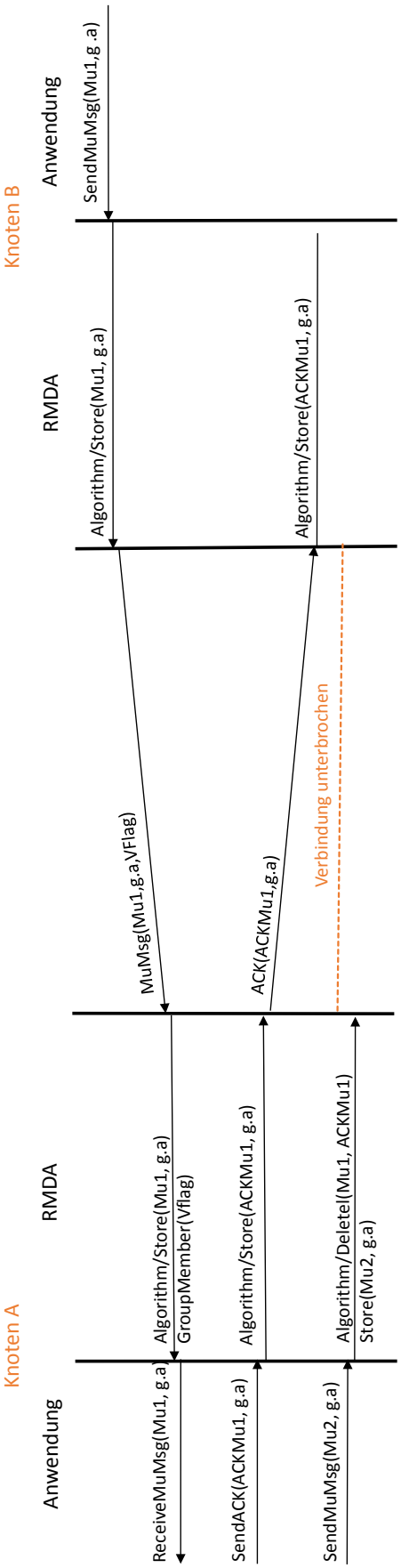

Abbildung 4.12: Weg-Zeit-Diagramm Multicastnachrichtenübertragung mit RMDA [Begerow u. a., 2015]

Ist nicht genug Speicher vorhanden, wird mit Hilfe des RMDA-Algorithmus entschieden, welche Nachrichten gelöscht werden (`Algorithm/Store`). Knoten B speichert und leitet das *ACK* solange weiter, bis laut RMDA-Algorithmus dieses gelöscht wird. Die Anwendung von Knoten A erzeugt eine *MuMsg MU2* an die Gruppe *g.a.* Der RMDA-Algorithmus entscheidet, dass aus Speicherplatzmangel die *MuMsg MU2* mit dem zugehörigen *ACK* gelöscht wird (`Algorithm/Delete`). Anschließend kann *MuMsg MU2* gespeichert werden (`Store`).

Nach der konzeptionellen Darstellung des RMDA-Protokolls und mit dessen Modulen wird im folgenden Kapitel anhand von Simulationsergebnissen überprüft und nachgewiesen, dass das gewählte Konzept die gestellten Anforderungen erfüllt.

5 Validierung

Das RMDA-Protokoll ist speziell für zuverlässige Multicastübertragungen in Katastrophenszenarien entworfen worden. Dieses Protokoll zeigt bisher ausschließlich die technische Möglichkeit einer Übertragung ohne Sicherheitsmechanismen, um die RMDA im realen Anwendungsfall erweitert werden muss. Deshalb wurde RMDA vorerst nur simuliert und noch nicht implementiert.

5.1 ONE-Simulator

Der Opportunistic Networking Environment (ONE)-Simulator [Keränen u. a., 2009] ist ein speziell für DTNs entwickelter Netzwerksimulator. Er wurde unter der objektorientierten Programmiersprache Java entwickelt und ist unter der General Public License (GPL)v3-Lizenz veröffentlicht.

ONE bietet bereits eine Reihe von bekannten DTN-Routingprotokollen, wie z.B. das Spray-and-Wait [Spyropoulos u. a., 2005], das ProPHET [Lindgren u. a., 2012] und das Epidemic Routing Protocol [Vahdat und Becker, 2000]. Unterliegende Schichten werden verallgemeinert. Bekannte Bewegungsmodelle, wie z. B. Random Walk [Einstein, 2011], Random Waypoint [Yoon u. a., 2003] und Working Day Movement [Ekman u. a., 2008], sind integriert. Die Map-based Modelle beziehen ihre Konfigurationsdaten aus Dateien im Well Known Text (WKT)-Format. Diese Dateien können mit Hilfe von Geographic Information System (GIS)-Programmen, beispielsweise mit OpenJump [OpenJump, 2011], bearbeitet und erzeugt werden. Darüber hinaus bietet ONE eine Reihe von Berichten, die einfache Messungen der gewünschten Metriken, wie Verzögerung oder Auslieferungsgrad, ermöglichen.

Die eingebauten Routingprotokolle und Berichte sind ausschließlich für Unicastnachrichten ausgelegt. Deshalb wurde die Implementierung von zwei Multicastprotokollen,

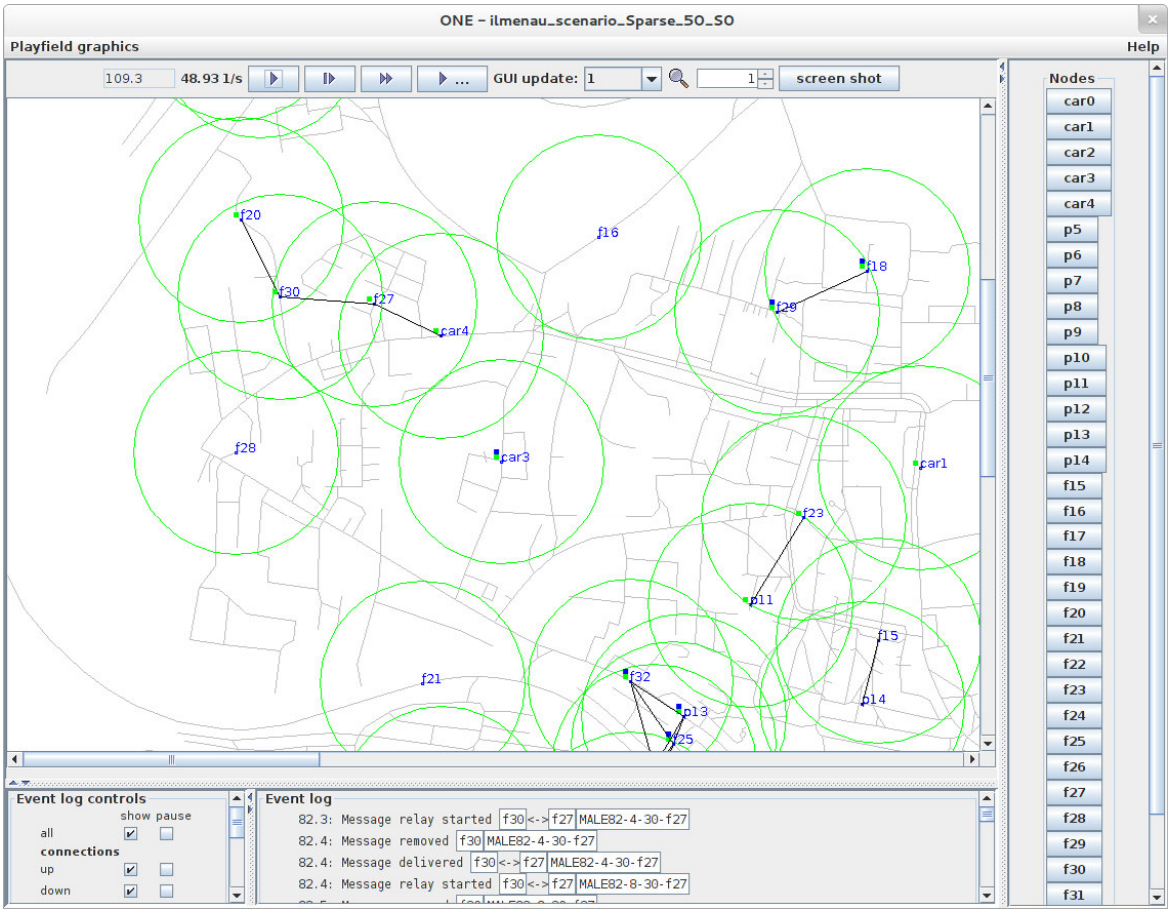

Abbildung 5.1: Screenshot ONE-Simulator

dem ECAM-Protokoll [Jin u. a., 2010] und dem BBR-Protokoll [Zhao u. a., 2005], notwendig. Näheres dazu im Abschnitt 5.3.6.

Parameter, wie Simulationsgebiet, Datenrate (Bytes pro Sekunde) und Knotengeschwindigkeit, können mit einer Konfigurationsdatei eingestellt werden. Außerdem bieten Ereignisdateien eine zeitgenaue Erstellung von Nachrichten und anderen Ereignissen, wie z. B. den Beitritt eines Knotens zu einer Gruppe. Tracedateien ermöglichen Pfadvorgaben für Knoten.

ONE bietet eine grafische Oberfläche und ist sehr bedienerfreundlich. Die Abbildung 5.1 zeigt eine Aufnahme einer Simulation, bei der die Karte des Ilmenauer Campus als Grundlage dient. ONE platziert in diesem Beispiel die Knoten nur auf den verfügbaren Wegen. Diese Knoten dürfen sich während der Simulation auch nur auf diesen Wegen bewegen. Die grünen Kreise kennzeichnen die Funkreichweite (TX range) der jeweiligen Knoten. Diese Reichweite ist in ONE idealisiert. Streuung und Abschattung durch

Gebäude oder andere Umwelteinflüsse werden nicht berücksichtigt. Zur besseren Übersicht sind Knoten, welche in Kontakt miteinander stehen, mit einer schwarzen Linie verbunden. Jeder Knoten ist während der Simulation einzeln anwählbar. So besteht die Möglichkeit den Nachrichtenfluss zu überprüfen.

Alle Simulationen, die in dieser Arbeit durchgeführt wurden, wurden mit ONE erstellt.

5.2 Umsetzung von RMDA

Für die Umsetzung des Konzeptes des RMDA-Protokolls aus dem Kapitel 4 wurde der nachstehende Pseudocode aufgeteilt in Algorithmus 6 bis Algorithmus 9 in den ONE-Simulator integriert und getestet. Auf Grundlage dieses Pseudocodes wurden alle Simulationen aus dem Unterkapitel 5.3 durchgeführt.

Um eine ankommende Nachricht im Nachrichtenspeicher ablegen zu können, wird zuerst geprüft, ob genügend Speicherplatz für diese Nachricht vorhanden ist. Ist dies nicht der Fall, wird Algorithmus 6 aufgerufen, der die Nachrichten, die gelöscht werden sollen, zurückgibt. Dieser Algorithmus ruft dafür weitere Funktionen, Algorihtmus 7, Algorithmus 8 sowie Algorithmus 9, auf. Auf diese Algorithmen wird im Folgenden noch detailliert eingegangen.

Zu Beginn werden alle Nachrichten, die an den Algorithmus 6 übergeben worden sind, nach Nachrichtentyp sortiert. Wird eine *MngMsg* gefunden, wird diese in die Liste der zu löschenden Nachrichten übernommen. Allerdings dürften keine *MngMsg* vorhanden sein, da diese normalerweise nach Übergabe an die höhere Schicht aus dem Speicher gelöscht wurden.

Nach der Nachrichtenaufteilung wird im zweiten Schritt der Algorithmus 7 aufgerufen, der die *MuMsg* mit ihren zugehörigen *ACKs* zurück gibt, die gelöscht werden können. Wurden keine Nachrichten gefunden, wird im Schritt drei der Speicherplatzanteil der *MuMsgs* (mit *ACKs*) ermittelt.

Ist die Kapazität für *MuMsgs* erreicht, wird Schritt drei ausgeführt. Algorithmus 8 ermittelt eine *MuMsg* mit zugehörigen *ACKs*, die prozentual die meisten *ACKs* zu den geschätzten Empfängern erhalten hat.

Algorithmus 6 Finde zu löschende Nachrichten

Require: Msg = alle Nachrichten, Managementlisten
Ensure: Del = alle Nachrichten die gelöscht werden sollen

```
1: for Msg do
2:     if msg[i].typ == null then                                   ▷ Schritt 1
3:         Nachrichtengröße zu Speicherverbrauch Unicast hinzufügen (bufferUni)
4:         Hinzufügen zu Unicastnachrichten (UMsg)
5:     end if
6:     if msg[i].typ == ACK then
7:         Nachrichtengröße zu Speicherverbrauch ACKs hinzufügen (bufferAcK)
8:         Hinzufügen zu ACKs (ACKs)
9:     end if
10:    if msg[i].typ == multicast then
11:        Nachrichtengröße zu Speicherverbrauch Multicast hinzufügen (bufferMul)
12:        Hinzufügen zu Multicastnachrichten (MulMsg)
13:    end if
14:    if msg[i].typ == management then
15:        Hinzufügen zu Del
16:    end if
17: end for
18: if del.size == null then
19:    Algorithmus 7                                                ▷ Schritt 2
20:    if del.size == null then      ▷ Wenn keine zu löschende Nachricht gefunden
21:        bufferAll=bufferMult + bufferACK + bufferUni
22:        maxBufferMul=ProportionMul*bufferAll/100;                ▷ Schritt 3
23:        if bufferMul + bufferACK > maxBufferMul then
24:            Algorithmus 8                                        ▷ Schritt 4
25:        else
26:            Algorithmus 9 (UMsg)                                 ▷ Schritt 5
27:        end if
28:        if del.size == null then
29:            Algorithmus 9 (msg)                                  ▷ Backupfunktion
30:        end if
31:    end if
32: end if
```

Schritt vier wird dann ausgeführt, wenn die Kapazität für *MuMsg* nicht erreicht ist. Dann wird die älteste *UMsg*, mit dem Algorithmus 9 ermittelt und gelöscht.

Algorithmus 7 RMDA-Algorithmus

Require: MulMsg, ACKs, Managementlisten
Ensure: Del = alle MulMsgs mit ACKs die gelöscht werden sollen

```
1: for Msg do
2:     for Gruppen do                          ▷ Für jede Gruppe aus GroupList
3:         if Zielgruppe aus Msg == aktuelle Gruppe aus GroupList then
4:             for Gruppenmitglieder do
5:                 MeanAll = Mittelwert über alle historischen Listen
6:                 MeanTeil = Mittelwert ab Startliste
7:             end for
8:             MemNeares = Mittelwert der Liste mit Zeitstempel
9:             Trend = MeanAll - MeanTeil
10:            Koeffizient = MeanAll *(100 - Level Erhaltene ACKs)/100
11:            if Trend > 0 then
12:                Varianz = Trend + Koeffizient
13:            else
14:                Varianz = Koeffizient
15:            end if
16:            for ACKs do
17:                if ACKs to Msg[i].ID then
18:                    CountACKs ++                ▷ Summiere ACKs
19:                end if
20:            end for
21:            if CountACKS >= (MemNeares - Varianz - 1) then
22:                Del = Del + Multicastnachricht mit zugehörigen ACKs
23:            end if
24:        end if
25:    end for
26: end for
```

Konnte trotzdem keine Nachricht gefunden werden, wird eine Backupfunktion ausgeführt. Dabei dieser wird die älteste, vom Typ unabhängige, Nachricht gelöscht. Der Algorithmus 7 zeigt die grundlegende Arbeitsweise des RMDA-Algorithmus, dessen detaillierte Erläuterung schon aus Abschnitt 4.4.5 bekannt ist. Dem Algorithmus müssen alle *MuMsgs* sowie *ACKs* übergeben werden. Des Weiteren müssen die Gruppenlisten und deren Mitglieder bekannt sein. Für alle *MuMsgs* wird dann geprüft, ob diese mit ihren zugehörigen *ACKs* gelöscht werden können, in diesem Fall wird eine Liste der zu löschenden Nachrichten zurückgegeben. Für die Berechnung wird der Mittelwert über alle Mitglieder gebildet. Dies geschieht anhand der zur Gruppe gehörenden Mitglieder-

informationen *MemberList*. Zusätzlich wird ein weiterer Mittelwert über die Mitglieder gebildet, von dem Zeitpunkt beginnend, der laut Zeitstempel am nächsten des Erstellungsdatums (`CreationTime`) der *MuMsg* ist. Daraus wird anschließend der aktuelle Trend ermittelt.

Algorithmus 8 getMulticastMessage

```
Require: MulMsg, ACKs, Managementlisten
Ensure: Del = alle MulMsgs mit ACKs die gelöscht werden sollen
 1: for Msg do
 2:     for Gruppen do                          ▷ Für jede Gruppe aus GroupList
 3:         if Zielgruppe aus Msg == aktuelle Gruppe aus GroupList then
 4:             for Gruppenmitglieder do
 5:                 MeanAll = Mittelwert über alle Listen
 6:                 MeanTeil = Mittelwert ab Startliste
 7:             end for
 8:             MemNeares = Mittelwert der Liste mit Zeitstempel
 9:             Trend = MeanAll - MeanTeil
10:             Koeffizient = MeanAll *(100 - Level Erhaltene ACKs)/100
11:             if Trend > 0 then
12:                 Varianz = Trend + Koeffizient
13:             else
14:                 Varianz = Koeffizient
15:             end if
16:             for ACKs do
17:                 if ACKs to Msg[i].ID then
18:                     CountACKs ++                ▷ Summiere ACKs
19:                 end if
20:             end for
21:             Prozent = CountACKS * 100/( MemNeares - Varianz);
22:             if Prozent >=ProzentACKs or ID==null then
23:                 ID = msg[i].ID
24:                 ProzentACKs = Prozent
25:             end if
26:         end if
27:     end for
28: end for
29: if ID!=null then
30:     Del = Del + Multicastnachricht mit zugehörigen ACKs
31: end if
```

Algorithmus 8 ermittelt diejenige *MuMsg* mit ihren zugehörigen *ACKs*, die nach Schätzung der Gruppenmitgliedsanzahl den höchsten Prozentsatz an *ACKs* erhalten hat und gibt diese an die übergeordnete Funktion zurück. Diese Funktion arbeitet ähnlich wie

Algorithmus 7. Allerdings wird nur die *MuMsg* mit dem höchsten Prozentsatz an *ACKs* ermittelt.

Algorithmus 9 ermittelt die älteste zu löschende *Msg* anhand der übergebenen Nachrichtenliste. Dabei wird nicht nach dem Typ der Nachricht unterschieden, lediglich der Zeitpunkt des Erzeugens der Nachricht wird betrachtet.

Algorithmus 9 getOldestMessages

```
Require: Msg
Ensure: Del = älteste Msg
 1: for Msg do
 2:     if Älteste Msg == Null then
 3:         Älteste Msg = Msg
 4:     else
 5:         if Älteste Msg.Erstellungszeitpunkt > Msg.Erstellungszeitpunkt) then
 6:             Älteste Msg = Msg
 7:         end if
 8:     end if
 9: end for
```

5.3 Simulationesbasierte Untersuchung

5.3.1 Gruppenlistenaustausch

Der Gruppenlistenaustausch ist eine wichtige Voraussetzung für das RMDA-Protokoll. Die Abbildung 5.2 zeigt den unterschiedlichen Einfluss der Knotendichte auf den Gruppenlistenaustausch. Die folgenden Einstellungen und Ergebnisse wurden in [Begerow u. a., 2014a] erstmals vorgestellt.

Nachgestellt soll im folgenden Szenario eine Katastrophe auf dem Ilmenauer Campus der Technischen Universität Ilmenau (Deutschland) werden. Dafür wurden Karteninformationen aus OpenSteetMap [OpenStreetMap-Mitwirkende, 2014] exportiert. Aus den so gewonnenen Daten wurden die Knotenpfade modelliert. In einem vorgegeben Gebiet (siehe Tabelle 5.1) bewegt sich ein Rettungsteam (Knoten) zufällig auf den vorgegebenen Fußwegen. Diese Modellierung zeigt ansatzweise, dass realistische Szenarien für Simulationen notwendig und weiterentwickelt werden müssen, wie in der Ausarbeitung [Krug u. a., 2014b] bestätigt wird.

Tabelle 5.1: Simulationsparameter für Gruppenverwaltung [Begerow u. a., 2014a]

Parameter	Werte
Simulationsgebiet	3000 m x 2000 m
Simulationszeit	7200 s
Reichweite (Tx range)	100 m
Knotenanzahl (Spärliches Szenario)	35
Knotenanzahl (Dichtes Szenario)	170
Gruppenmitglieder (Spärliches Szenario)	10
Gruppenmitglieder (Dichtes Szenario)	50
Knotengeschwindigkeit	0 .. 5.4 $^{km}/_{h}$

Diese Simulation wurde mit einer unterschiedlichen Anzahl von Knoten durchgeführt. Dabei stellt das „spärliche Szenario“ eine Situation dar, bei der sich nur wenige Knoten in einem Gebiet aufhalten und somit, durch die begrenzte Sendereichweite, nur wenig Kontakt miteinander haben, um Daten und *MgmtMsgs* auszutauschen. Für das „spärliche Szenario“ wurde eine Knotenanzahl von 35 Knoten angenommen, wobei von diesen Knoten, 10 ausgewählte Knoten, Gruppenmitglieder werden sollen. Das „dichte Szenario“ beschreibt eine Situation, bei der sich viele Knoten in einem bestimmten Gebiet aufhalten. Durch den häufigen Kontakt zu anderen Knoten können *MgmtMsgs* als auch Daten häufiger ausgetauscht werden. Für das „dichte Szenario“ bewegten sich 175 Knoten auf der gleichen Grundfläche, wie bei der Simulation im „spärliche Szenario“, von denen 50 der Gruppe beitreten sollen. Fünf Sekunden nach dem Start der Simulation wird die Gruppe durch einen vordefinierten Knoten eingerichtet. Erhalten die Knoten, die ausgewählt wurden der Gruppe beizutreten, durch eine *MngMsg* die Kenntnis, dass diese Gruppe existiert, treten sie dieser Gruppe bei. Das Beitreten zur Gruppe wird wiederum durch eine *MngMsg*, über den Kontakt mit anderen Knoten, weitergeleitet.

Abbildung 5.2 zeigt, wie viel Zeit benötigt wird, bis jedes Gruppenmitglied alle Gruppenmitglieder kennt. Die Simulation selbst wurde 10 mal mit unterschiedlichen Bewegungsmustern durchgeführt und gemittelt dargestellt. Die Abweichungen reflektieren die Unsicherheit des lokalen Wissens in einzelnen Knoten, weshalb der mögliche untere abweichende Bereich im RMDA-Algorithmus, bekannt als *range*-Paramenter, Formel 4.7, einbezogen wurde. Diese Abweichung ist am Anfang des Austauschs von *MngMsgs* höher als in einem eingeschwungenen Zustand.

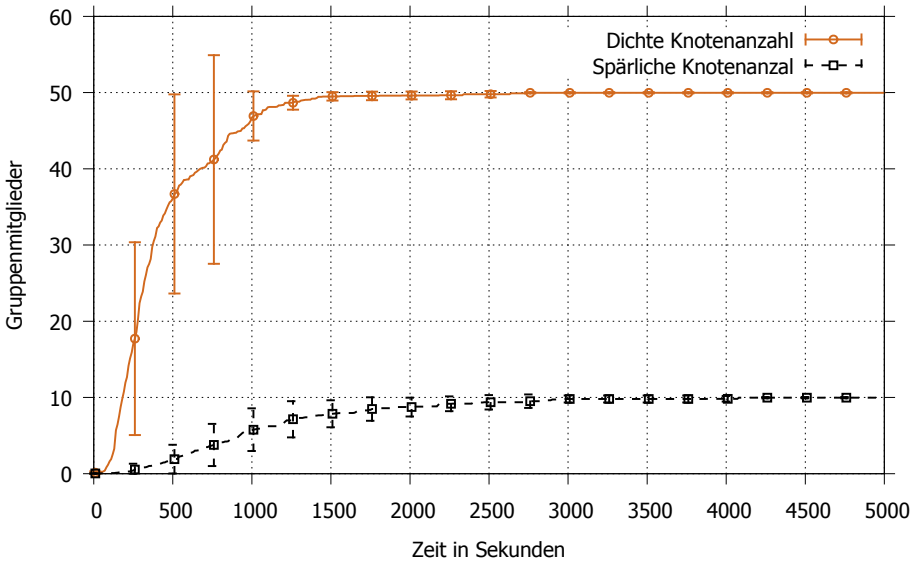

Abbildung 5.2: Einfluss Knotendichte beim Gruppenlistenaustausch [Begerow u. a., 2014a]

Im Vergleich zu dem „spärlichen Szenario“ ist der Listenaustausch im „dichten Szenario“ deutlich schneller, obwohl die Gruppe viel mehr Mitglieder aufweist. Der Listenaustausch erfolgt deshalb zügiger, weil die Wahrscheinlichkeit einen Knoten zu treffen und somit seine Informationen zu verteilen, viel höher ist. Daher hängt die Dauer des Listenaustausches direkt von der Kontakthäufigkeit ab.

In Abbildung 5.2 wurde im „spärlichen Szenario“ nach 3000 Sekunden die Erstellung einer neuen Gruppe bekannt gemacht und nahezu 100 Prozent der Knoten informiert, dass 10 Gruppenmitglieder dieser beigetreten sind, wohingegen im „dichten Szenario“ schon nach 1500 Sekunden die Gruppeninformationen ausgetauscht waren. Ist die Gruppenmitgliedsanzahl schon zu Beginn der Mission relativ konstant, kann ein stabiler Zustand beobachtet werden. Deshalb kann angenommen werden, dass nach einer Einschwingphase die Schätzung der Gruppenmitgliedszahl ziemlich genau ist und einer quasi globalen Sicht entspricht.

5.3.2 Einfluss Speichergröße

Trotz der heutzutage kostengünstigen Speicherplätze ist es in Katastrophenszenarien trotzdem notwendig, Geräte mit möglichst kleinen Speichern auszurüsten. Kleine Speicher ermöglichen schnelleres Abspeichern, Verarbeiten sowie Abrufen von Daten.

Hoch mobile MANETs sowie DTNs sind gekennzeichnet durch kurze Kontaktzeiten der Knoten untereinander.

Die Untersuchung, welchen Einfluss die Speichergröße auf die Speicherverwaltung hat, wurde in [Begerow u. a., 2014a] erstmals, durchgeführt. Die dort genutzte Tracedatei wurde mit Hilfe von BonnMotion Aschenbruck u. a. [2010] erzeugt. Die Abbildung 5.3 hingegen zeigt den Mittelwert von fünf Simulationen, die ausschließlich mit dem Random-Waypoint-Generierung aus dem ONE-Simulator erstellt wurden.

Für die Simulation wurden 20 Knoten dem Simulationsgebiet zugeteilt. Es wurde eine Gruppe erzeugt, und jeder der Knoten tritt dieser Gruppe bei. Zwischen der fünfzigsten und sechzigsten Sekunde der Simulationen sendet jeder Knoten genau eine *MuMsg* an diese Gruppe. In Tabelle 5.2 sind alle relevanten Simulationsparameter zu sehen.

Tabelle 5.2: Simulationsparameter für Einfluss der Speichergröße

Parameter	Werte
Simulationsgebiet	340 m x 340 m
Simulaitonszeit	300 s
Reichweite (Tx range)	170 m
Datenrate	250 $^{kB}/_{s}$
Knotengeschwindigkeit	1 $^{m}/_{s}$
Nachrichtengröße	500 kB
Knotenanzahl	20

Wie im Beispiel aus Abbildung 5.3 zu erkennen, erhöht sich die Zustellrate mit steigender Speicherplatzgröße. Ähnlich den Ergebnissen aus [Begerow u. a., 2014a], wird die Zustellrate von ca. 100 Prozent, ab einer Speicherplatzgröße von 4 MB erreicht. Die Abweichungen um den Mittelwert entstehen durch die unterschiedlichen Bewegungsmuster.

Ändert sich das Nachrichtenaufkommen oder kommen weitere Gruppenmitlieder hinzu, wird sich auch die 100 Prozentmarke dementsprechend verschieben. D. h. je mehr Nachrichten versendet werden, desto mehr Speicher wird benötigt, um eine Zustellrate von 100 Prozent zu erreichen. In der Abbildung 5.3 wird deutlich, dass das RMDA-Protokoll eine Zustellrate von 100 Prozent erreichen kann, wenn ausreichend Nachrichtenspeicher vorhanden ist.

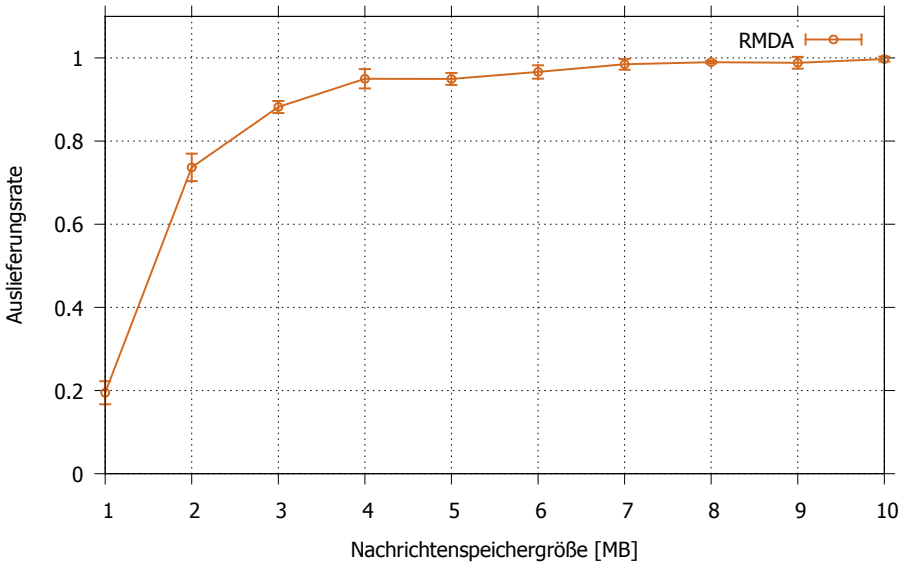

Abbildung 5.3: Einfluss Speichergröße

5.3.3 Levelermittlung

Die im Abschnitt 4.4.4 vorgestellten drei Level wurden experimentell ermittelt. Dabei wurde untersucht, wie der Speicherplatzanteil Einfluss auf das Auslieferverhalten von *MuMsgs* und *UMsgs* hat.

Tabelle 5.3: Simulationsparameter für die Ermittlung des optimalen Speichers je Level

Parameter	Value
Simulationsgebiet	500 m x 500 m
Simulationszeit	2000 s
Reichweite (Tx range)	100 m
Datenrate	250 $^{kB}/_{s}$
Knotengeschwindigkeit	1 $^{m}/_{s}$
Nachrichtengröße Unicast	250 kB
Nachrichtengröße Multicast	250 kB

Die Knoten wurden auf drei Gruppen aufgeteilt. Die Knotenanzahl selbst wurde von 10 bis 45 Knoten in einer Schrittweite von fünf Knoten festgelegt. Es sendet jeder Knoten eine *MuMsgs* an die Gruppe. Im gleichen Zeitraum senden fünf andere Knoten ein *UMsgs* an einen zufälligen Empfänger. Das Verhältnis der gesendeten *MuMsgs* zu gesendeten *UMsgs* beträgt also 1 : 5.

Bei der heutigen Kommunikation zwischen Rettungskräften überwiegt der Multicastanteil, denn um Ressourcen zu sparen, werden *UMsgs* über Multicastgruppen gesendet. Beim TETRA-Standard wird nur ein Zeitschlitz für eine Gruppennachricht, statt zwei für Einzelruf benötigt. Zukünftig werden deshalb diese vermeintlichen *MuMsgs* auch als tatsächliche *UMsgs* verschickt werden.

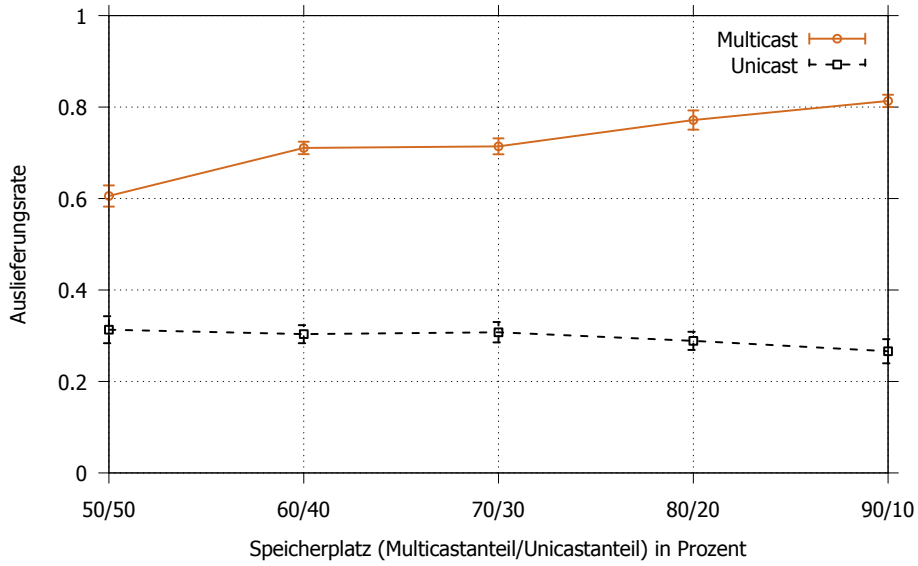

Abbildung 5.4: Ermittlung optimaler Speicherplatzanteil für Level 1

Für jedes Level wurde das vorgesehene Speicherplatzverhältnis für *MuMsgs* zu *UMsgs* unterschiedlich aufgegliedert. Begonnen wurde mit der hälftigen Aufteilung, d. h. 50 Prozent *MuMsgs* zu 50 Prozent *UMsgs*. Für jedes Level wurde dann der Multicastanteil um 10 Prozent erhöht, bzw. der Unicastanteil um 10 Prozent verringert. Dies wurde in 10 Prozent Schritten fortgeführt, bis ein Verhältnis von 90 Prozent Multicastanteil zu 10 Prozent Unicastanteil erreicht wurde.

Für die optimale Speicherermittlung wurden insgesamt 200 Simulationen für jedes Level durchgeführt. Es wurde ein Mittelwert über jeweils fünf Simulationen mit gleichem Speicheranteil pro Knotenanzahl gebildet. Alle weiteren Simulationsparameter sind aus Tabelle 5.3 zu entnehmen.

Level 1 hat die Anforderung eine möglichst hohe Zustellrate für *MuMsgs* zu erreichen. In Abbildungen 5.4 ist deutlich zu sehen, dass dies bei dem Speicherplatzverhältnis von 90 Prozent Multicastanteil zu 10 Prozent Unicastanteil erreicht wird. In diesem Beispiel wird so eine Auslieferungsrate der *MuMsgs* von 81 Prozent erreicht. *UMsgs* erreichen nur eine Auslieferungsrate von 27 Prozent.

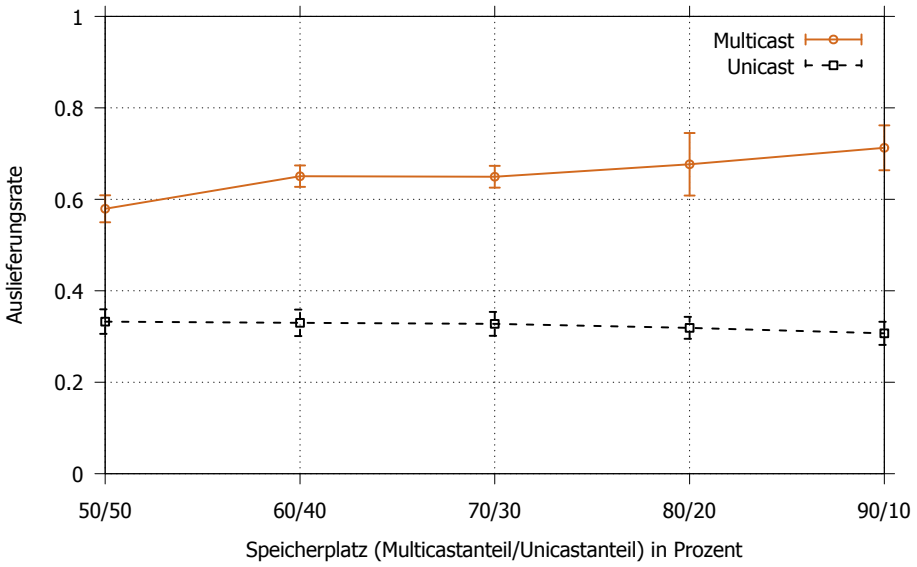

Abbildung 5.5: Ermittlung optimaler Speicherplatzanteil für Level 2

Für Level 2 (siehe Abbildung 5.5) liegt der Fokus der Auslieferung immer noch auf dem Multicastanteil. Die Auslieferungsrate von *UMsgs* sollte aber trotzdem gesteigert werden. Deshalb wurde sich für dieses Level für eine Speicherplatzverteilung von 80 Prozent Multicastanteil zu 20 Prozent Unicastanteil entschieden. Das Beispiel aus Abbildung 5.5 zeigt, dass bei dieser Verteilung die Auslieferungsrate der *MuMsgs* immer noch bei 68 Prozent liegt und der Unicastanteil sich auf 32 Prozent beläuft.

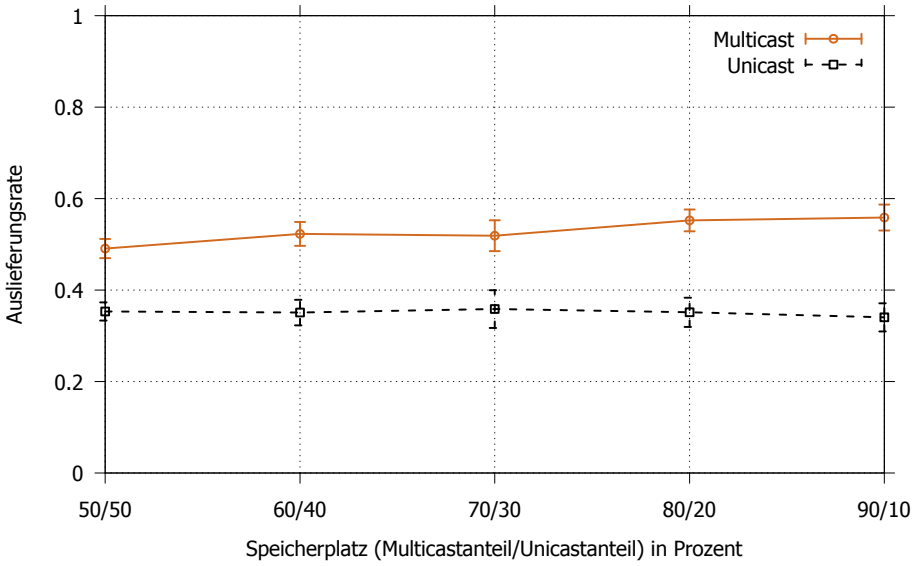

Abbildung 5.6: Ermittlung optimaler Speicherplatzanteil für Level 3

Da das RMDA-Protokoll das Hauptaugenmerk auf die zuverlässige Multicastübertragung legt und trotzdem ein höchst möglicher Anteil von *UMsgs* erreicht werden soll, wurde sich in Level 3 auf die Verteilung 70 Prozent Speicherplatz für *MuMsgs* zu 30 Prozent Speicherplatz für *UMsgs* entschieden. In Abbildung 5.5 zeigt das Beispiel eine Auslieferungsrate von 52 Prozent *MuMsgs* und die Auslieferungsrate von *UMsgs* liegt bei diesem Verhältnis bei 36 Prozent.

5.3.4 Einfluss Levelauswahl auf den Auslieferungsgrad

Um den Einfluss der verschiedenen Levels darzulegen, wurden Simulationen mit unterschiedlichen Mengen an Knoten in einem festen Gebiet durchgeführt. Vorgestellt wurden erste Ergebnisse, mit einer Knotendichte von bis zu 35 Knoten, in [Begerow u. a., 2015]. Im Folgenden wurde die Simulation bis zu einer Knotendichte von 45 Knoten erweitert.

Für diese Simulation wurde ein Gebiet von 500 m x 500 m festgelegt, in dem sich Knoten nach dem Random-Waypoint-Bewegungmodell fortbewegen und somit ein DTN erzeugen, wie es in Katastrophenfällen vorkommt. Für jedes Level gelten die selben Simulationsparameter, die in Tabelle 5.4 beschrieben sind.

Für die Simulation wurden drei Gruppen erzeugt. Jeder Knoten tritt nur einer Gruppe bei. Um optimale Vergleichsparameter zu schaffen, wurde eine externe Ereignisdatei generiert, welche *MuMsgs* und *UMsgs* in einem Verhältnis von 1 : 5 erzeugt. Jeder Knoten sendet eine *MuMsg* an seine Gruppe. Zeitgleich werden fünf weitere *UMsgs* generiert. Das bedeutet je mehr Knoten, desto mehr *MuMsgs* als auch *UMsgs* werden erzeugt.

Zu Beginn werden die Gruppeninformationen ausgetauscht. Die eigentliche Nachrichtenübertragung beginnt nach 1000 s, damit wird sichergestellt, dass alle Knoten die Managementlisten ausgetauscht haben. Dieses Vorgehen sichert ein eindeutigeres Ergebnis, denn Einflüsse, die auf fehlende Gruppeninformationen durch die unterschiedlichen Bewegungsmuster entstehen, werden so ausgeschlossen. In der Realität ist ein Warten auf den Austausch nicht erforderlich. Die Nachrichtengenerierung selbst erfolgt 100 s lang. Die Simulationsdauer wurde auf 2000 s festgelegt, wodurch den Nachrichten die Möglichkeit geben wird, auch bei unterbrochener Verbindung, die Empfänger zu erreichen.

Tabelle 5.4: Simulationsparameter für Levelauswahl

Parameter	Value
Simulationsgebiet	340 m x 340 m
Simulationszeit	2000 s
Reichweite (Tx range)	100 m
Datenrate	250 $^{kB}/_{s}$
Knotengeschwindigkeit	1 $^{m}/_{s}$
Nachrichtengröße Unicast	250 kB
Nachrichtengröße Multicast	250 kB
Knotenspeicher	2 MB

Wie die einzelnen Levels das Verhältnis der tatsächlichen Lieferung beeinflussen, wird in Abbildung 5.7 dargestellt. Es ist deutlich zu sehen, dass der Auslieferungsgrad bei zunehmender Anzahl der Knoten sinkt. Die Ursache dafür ist das schnellere Erreichen des Endes der Speicherkapazität des Nachrichtenspeichers, durch den erhöhten Nachrichtenfluss bei einer höheren Knotendichte. Das erfordert ein früheres Löschen von Nachrichten, auch wenn diese Nachrichten noch nicht alle Empfänger erreicht haben. Jedoch löscht jeder Knoten aus lokaler Sicht nach den Kriterien des RMDA-Protokolls verschiedene Nachrichten, weshalb nicht bei jedem Knoten die gleichen Nachrichten gelöscht werden. Daher besteht die Möglichkeit, dass eine gelöschte Nachricht von einem anderen Knoten an die noch fehlenden Empfänger ausgeliefert werden kann.

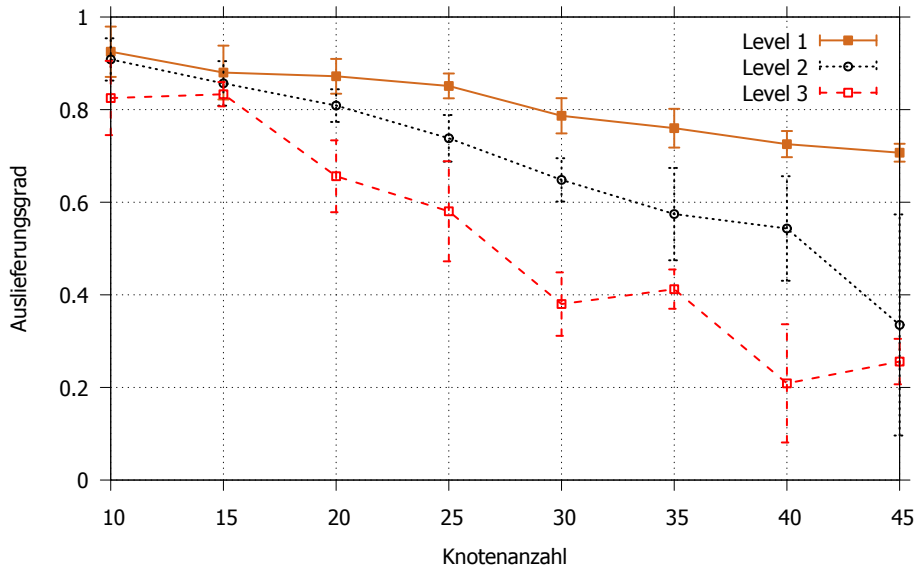

Abbildung 5.7: Einfluss der verschiedenen Level auf die Auslieferung

Abbildung 5.7 zeigt die unterschiedlichen Auslieferungsgrade bei den verschiedenen Leveln. Level 1 zeigt den besten Auslieferungsgrad. Level 2 wiederum ermöglicht einen besseren Auslieferungsgrad als Level 3. Das entspricht den Ergebnissen aus [Begerow u. a., 2015] und demzufolge dem RMDA-Konzept. Level 3 zeigt die größten Schwankungen im Auslieferungsgrad. Der Speicherplatzanteil für Multicastnachrichten ist bei diesem Level am niedrigsten und liegt für Multicastnachrichten bei 70 Prozent. Multicastnachrichten können auch schon bei 50 Prozent der zu erwartenden ACKs gelöscht werden. Die Knoten haben aber keine Kenntnis darüber, mit welchem Knoten sie als nächstes Kontakt haben. Dies kann dazu führen, wie beispielsweise bei der Knotenanzahl 15 zu sehen, dass ein sinkender Auslieferungsgrad erwartet wird, dennoch ein leichter Anstieg zu erkennen ist. In diesem Fall wurden zufällig die Nachrichten gelöscht, bei dem der Knoten weniger Kontakt zu potentiellen Empfängern hat. Das zeigt deutlich, dass der Auslieferungsgrad mit dem Kontaktverhalten zum richtigen Zeitpunkt und den richtigen Nachrichten in engem Zusammenhang steht.

5.3.5 VFlag Einfluss

Das VFlag in der *MuMsg* spielt eine entschiedene Rolle bei der Empfängerauswahl und demzufolge auf die Nachrichtenauslieferung. Um zu zeigen, dass das Konzept funktioniert, wurde eine einfache Simulation durchgeführt. Diese ist in Abbildung 5.8 zu sehen.

In der folgenden Simulation gehören 20 Knoten zu unterschiedlichen Zeitpunkten einer Gruppe an. Der Beitrittszeitraum der Knoten wurde in der Tabelle 5.6 dargestellt. Die Spalte Beitritt zeigt den Zeitpunkt, wann der Knoten Gruppenmitglied geworden ist. Die Spalte Austritt kennzeichnet den Zeitpunkt des Beendens der Gruppenmitgliedschaft. Enthält die Spalte Austritt den Wert -1, wurde beim Beitritt kein Austrittsdatum mitgegeben. Somit ist die Gruppenmitgliedschaft solange gültig, bis die Gruppe selbst beendet wird oder mit Hilfe der Funktion `DeleteGroupmebership` ein Austrittszeitpunkt gesetzt wird. Es wurden der einfacher halber nur fünf *MuMsgs* zu unterschiedlichen Zeitpunkten von unterschiedlichen Knoten mit Hilfe einer Ereignisdatei erzeugt. Die Größe der *MuMsgs* wurde extra klein gewählt (5 kB), damit ein frühzeitiges Löschen durch die begrenzte Speicherkapazität des Nachrichtenspeichers verhindert wird. Die Nachrichtengenerierung ist in Tabelle 5.7 dargestellt. Die Simulationszeit beträgt 2000 Sekunden. Alle weiteren Simulationsparameter sind in Tabelle 5.5 aufgeführt.

Tabelle 5.5: Simulationsparameter für VFlag Einfluss

Parameter	Value
Simulationsgebiet	340 m x 340 m
Simulationszeit	2000 s
Reichweite (Tx range)	100 m
Datenrate	250 $^{kB}/_{s}$
Knotengeschwindigkeit	1 $^{m}/_{s}$
Nachrichtengröße Multicast	5 kB
Knotenspeicher	2 MB

Wie schon erwähnt, ist die Gruppenzugehörigkeit der einzelnen Knoten in Tabelle 5.6 dargestellt. Für jeden Knoten ist die Beitrittszeit zur Gruppe sowie der Austrittszeitpunkt in Sekunden aufgeführt. Ist der Austrittszeitpunkt -1, ist kein expliziter Austritt definiert und somit ist die Gruppenzugehörigkeit unendlich gültig.

Tabelle 5.6: Gruppenzugehörigkeit

Knoten	Beitritt	Ausstritt
0	1	-1
1	1000	-1
2	1000	-1
3	1000	-1
4	1000	-1
5	1000	1100
6	1000	1100
7	1000	1100
8	1000	1100
9	1000	1100
10	1100	1200
11	1100	1200
12	1100	1200
13	1100	1200
14	1100	1200
15	1100	1200
16	1100	1200
17	1100	1200
18	1100	1200
19	1100	1200

In Tabelle 5.7 sind alle *MuMsgs* aufgelistet. Die erste Spalte kennzeichnet die eindeutige Kennung der *MuMsg*, die zweite Spalte enthält die Kennung des Quellknotens (vereinfacht dargestellt). Die letzte Spalte gibt den Zeitpunkt der Nachrichtengenerierung in Sekunden an.

Tabelle 5.7: Multicastnachrichten

ID	Quellknoten	Erstellungszeit
MU1	0	1001
MU2	1	1051
MU3	2	1099
MU4	3	1101
MU5	4	1201

Wird zum Beispiel die *MuMsg MU3* betrachtet, liegt der Zeitpunkt (laut Tabelle 5.7) bei 1099 Sekunden. So müssten bei `VFlag` *01* (gültige Gruppenmitglieder, sind Gruppenmitglieder beim Erstellen der Nachricht) die Knoten 0 bis 9 (laut Tabelle 5.6) sein. Da der Knoten 2 aber die Nachricht generiert hat, ist dieser kein Empfänger der Nachricht. Somit beträgt die Gesamtempfängeranzahl neun Knoten, in Abbildung 5.8 ist genau diese Anzahl aufgezeigt.

Wird das `VFlag` von *MU3* auf *00* gesetzt (gültige Gruppenmitglieder sind beim Erhalt der Nachricht Gruppenmitglied), ist es laut Tabelle nicht ersichtlich, zu welchem Zeitpunkt die Knoten die Nachricht erhalten haben. Knoten 0, 1, 3 und 4 sind auf jeden Fall Gruppenmitglied bei Nachrichtenerhalt. Die Wahrscheinlichkeit, dass Knoten 5 bis Knoten 9 die Nachricht innerhalb einer Sekunde erhalten, ist relativ gering. Die Wahrscheinlichkeit, dass Knoten 10 bis Knoten 19 bei Erhalt der Nachricht Gruppenmitglied sind, ist sehr hoch. Deshalb ist der Wert aus Abbildung 5.8 mit elf Knoten ein akzeptabler Wert.

Wählt man bei *MU3* das `VFlag` *10* (gültiges Gruppenmitglied, wenn der Knoten beim Erhalt sowie des Erstellens der Nachricht Gruppenmitglied ist) sinkt die Empfängeranzahl rapide. In Abbildung 5.8 liegt der Wert daher bei sechs Knoten. Schaut man in die Tabelle 5.6, ist zu sehen, dass nur die Knoten 0, 1, 3 und 4 (Knoten 2 ist Sender) sichere Gruppenmitglieder sind. Weitere Empfänger können nur noch Knoten zwischen Knoten 5 und Knoten 9 sein. Die Knoten 10 bis 19 treten erst nach der Generierung der *MuMsg* bei.

Abbildung 5.8 zeigt darüber hinaus, dass alle Knoten, außer der Senderknoten 2 bei *MU3*, die Nachricht erhalten.

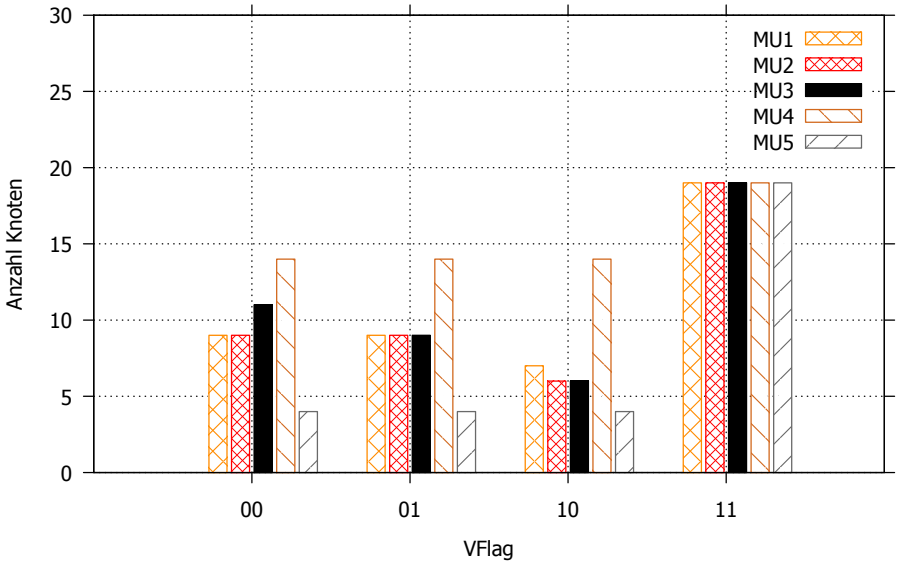

Abbildung 5.8: Einfluss des VFlags auf die Auslieferung

Das Beispiel zeigt deutlich, dass bei gleicher Knotendichte, bei gleicher Nachrichtengenerierung mit unterschiedlichem `VFlag` unterschiedliche Auslieferungsergebnisse erzeugt werden. Dies entspricht dem Konzept der Empfängeridentifikation aus Abschnitt 4.4.2.

5.3.6 Protokollvergleich

Der folgende Abschnitt zeigt die Leistung der unterschiedlichen Ansätze in Bezug auf die Auslieferungsrate.

5.3.6.1 Protokollvergleich nur Multicastnachrichten

Der erste Vergleich zwischen Multicastprotokollen basiert auf reiner Multicastübertragung. Dafür wurden das RMDA-Protokoll mit dem BBR-Protokoll [Zhao u. a., 2005] und dem ECAM-Protokoll [Jin u. a., 2010] verglichen. Durch die Annahme, dass durch flutenbasierte Ansätze eine besonders hohe Auslieferungsrate erzielt werden, wurde eine einfache Multicast Epidemic (MCEpidemic) Variante (BBR-Protokoll [Zhao u. a., 2005]) für den Vergleich mit dem RMDA-Protokoll ausgewählt.

Das ECAM-Protokoll ist ein Speicherverwaltungsprotokoll (siehe Abschnitt 3.3.2) und wurde deshalb als zweites Vergleichsprotokoll ausgesucht.

Tabelle 5.8: Simulationsparameter für Protokollvergleich

Parameter	Value
Simulationsgebiet	500 m x 500 m
Simulationszeit	2000 s
Reichweite (Tx range)	100 m
Datenrate	250 $^{kB}/_{s}$
Knotengeschwindigkeit	1 $^{m}/_{s}$
Nachrichtengröße Multicast	500 kB
Knotenspeicher	2 MB

Die Knotendichte für alle Simulationen variierte von 10 bis 55 Knoten in Schritten von fünf Knoten. Auch hier wurde das Random-Waypoint Bewegungsmodell angewendet. Es wurden für jede Knotendichte mehrere Simulationen durchgeführt und gemittelt. Alle wichtigen Simulationsparameter sind in Tabelle 5.8 aufgelistet.

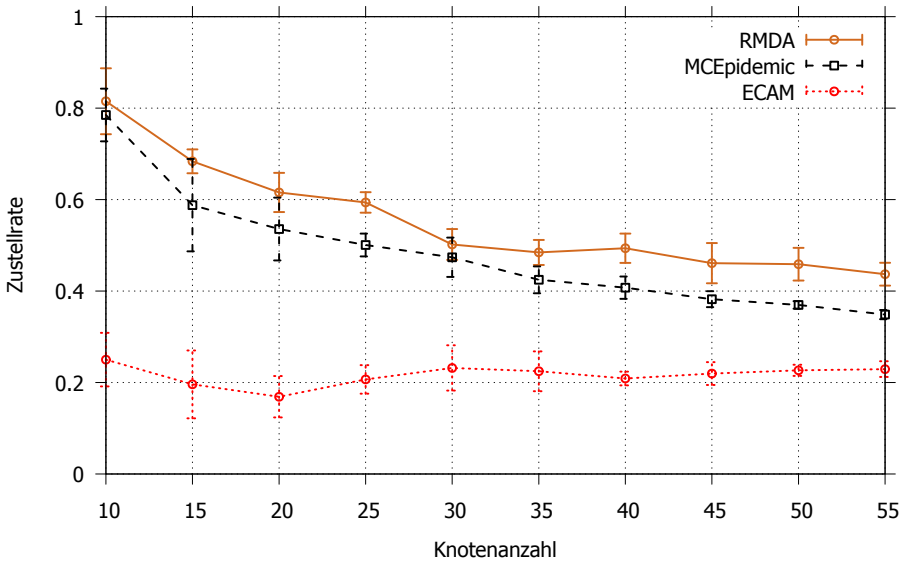

Abbildung 5.9: Protokollvergleich der Auslieferungsrate (2 Gruppen)

Das RMDA-Protokoll arbeitet auf Level 1, um die höchstmögliche Auslieferrate zu erzielen. Da keine Unicastnachrichten vorhanden sind, steht der gesamte Nachrichtenspeicher den Multicastnachrichten zur Verfügung.

ECAM verwendet die gleichen Werte wie von den Autoren für ihre Simulationen in [Jin u. a., 2010] ausgewählt wurden. Daher wurde der ECAM-Schwellwert auf sechs festgelegt und der ECAM-spezifische Hop-Zähler wurde manuell nach der Berechnungsbeschreibung von [Jin u. a., 2010] ermittelt, um die Anzahl der nicht-Gruppenmitglieder zu reflektieren. Die selben Werte wurden auch schon in [Begerow u. a., 2015] verwendet.

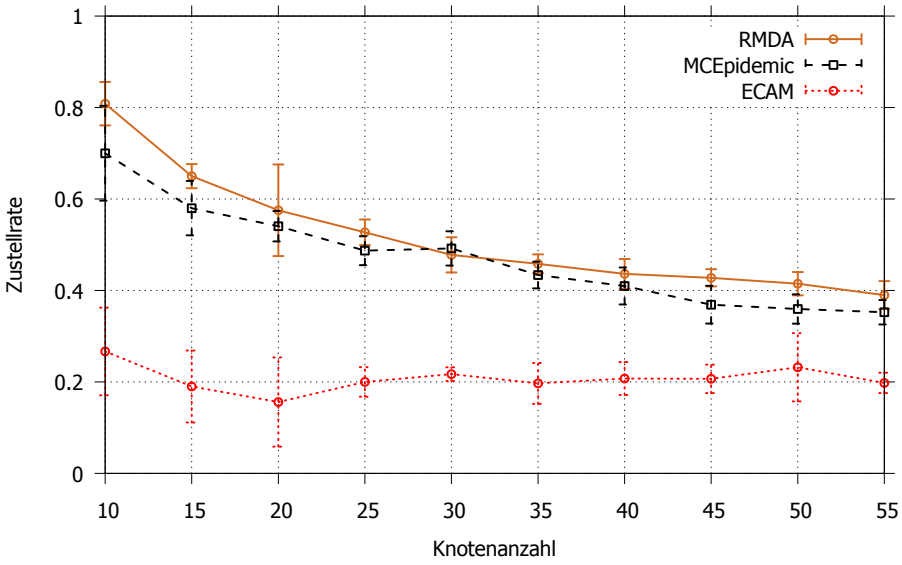

Abbildung 5.10: Protokollvergleich der Auslieferungsrate (3 Gruppen)

Nach erfolgten Gruppenlistenaustausch sendet jeder Knoten nur eine *MuMsg* an seine Gruppe. Die Simulationszeit von 2000 Sekunden erhöht die Chance auf die Zustellung der Nachrichten auch an die Knoten, die sich während der Nachrichtengenerierung außer Reichweite befunden haben.

Abbildung 5.9 zeigt die Multicastnachrichtenübermittlung, wenn die Knoten in zwei Gruppen aufgeteilt sind. Abbildung 5.10 weist die Knoten auf, die auf drei Gruppen verteilt sind. Der allgemeine Trend ist in beiden Abbildungen identisch. Das Auslieferverhältnis ist beim RMDA-Protokoll als auch beim MCEpidemic-Protokoll bei erhöhter Knotendichte also bei erhöhtem Datenverkehr rückläufig. Trotz zusätzlicher ACKs beim RMDA-Protokoll und dem damit verbundenen zusätzlich benötigten Nachrichtenspeicher, zeigt dieses Protokoll die besten Ergebnisse. Gleichzeitig zeigt das ECAM-Protokoll einen relativ konstanten niedrigen Auslieferungsgrad um die 20 %.

Bei reiner Multicastnachrichtenübertragung verringert sich der Abstand des Auslieferungsgrades zwischen RMDA und MCEpidemic bei erhöhter Gruppenanzahl. Jedoch ist eine reine Multicastübertragung nicht realistisch. Es wurden weitere Simulationen kombiniert mit *UMsgs* im Unterabschnitt 5.3.6.2 durchgeführt.

5.3.6.2 Protokollvergleich Multicastnachrichten und Unicastnachrichten

Die Kommunikation in Katastrophenszenarien ist gekennzeichnet durch Multicastnachrichtenübertragung und gleichzeitiger Übertragung von Unicastnachrichten. Die Knoten werden in fünf Gruppen aufgeteilt. Die Knotenanzahl selbst wurde von 15 bis 45 Knoten in einer Schrittweite von 5 Knoten festgelegt. Es generiert jeder Knoten eine *MuMsg* an die Gruppe. Im gleichen Zeitraum senden fünf andere Knoten ein *UMsg* an einen zufälligen Knoten. Das Verhältnis der gesendeten *MuMsgs* zu gesendeten *UMsgs* beläuft sich auf 1 : 5. Tabelle 5.9 zeigt die wichtigsten Simulationsparameter. Auch hier wurde das Random-Waypoint Bewegungsmodell angewendet.

Tabelle 5.9: Simulationsparameter Multicast gepaart mit Unicast

Parameter	Value
Simulationsgebiet	500 m x 500 m
Simulationszeit	2000 s
Reichweite (Tx range)	100 m
Datenrate	250 $^{kB}/_{s}$
Knotengeschwindigkeit	1 $^{m}/_{s}$
Nachrichtengröße Unicast	250 kB
Nachrichtengröße Multicast	250 kB

Der Protokollvergleich wurde nur zwischen dem MCEpidemic-Protokoll und dem RMDA-Protokoll durchgeführt. Im ECAM-Protokoll ist eine Unicastübertragung nicht definiert.

In der Abbildung 5.11 ist deutlich zu sehen, dass das RMDA-Protokoll doppelt so viele Nachrichten wie MCEpidemic ausliefert. Auch hier ist der typische Abwärtstrend bei Erhöhung der Knotendichte sichtbar. Trotz erhöhter Gruppenanzahl ist eine deutlich bessere Auslieferung der Multicastnachrichten beim RMDA-Protokoll zu erkennen.

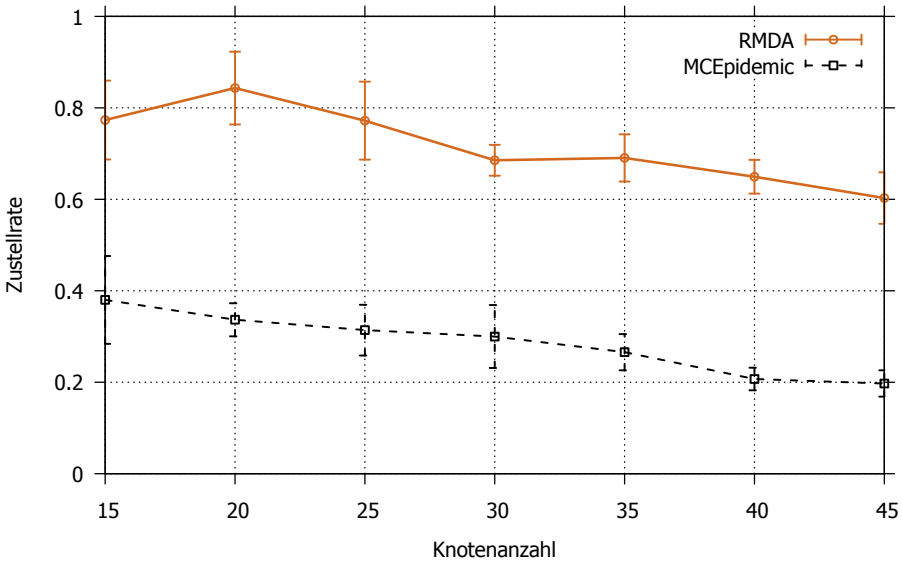

Abbildung 5.11: Protokollvergleich Multicastnachrichten und Unicastnachrichten (5 Gruppen)

Das Ergebnis der Simulationen zeigt, dass die Speicherverwaltungsstrategie zu besseren Ergebnissen führt. Damit ist einer höhere Zuverlässigkeit für Multicastübertragungen, insbesondere in Katastrophenszenarien gefordert, gewährleistet.

6 Zusammenfassung und Ausblick

Aus dieser Arbeit entstand ein Gruppenkommunikationsprotokoll speziell für verzögerungstolerante MANETs, respektive DTNs mit effizienter Speicherverwaltungsstrategie, welches den Anforderungen in Katastrophenszenarien entspricht. Diese neuartige Speicherverwaltungsstrategie führt zu einem verbesserten Auslieferungsgrad von Multicastnachrichten und erhöht somit die zuverlässige Datenübertragung.

Das RMDA-Protokoll gliedert sich in zwei Module, das Gruppenverwaltungsmodul und das Übertragungsmodul, welche sich dezentral in jedem RMDA-fähigen Knoten befinden.

Das Gruppenverwaltungsmodul stellt ein Gruppenmanagement zur Verfügung. Das Gruppenmanagement verwaltet Gruppen und deren Mitglieder. Zum Austausch von Gruppeninformationen werden Managementlisten mit Hilfe von Managementnachrichten ausgetauscht.

Auf Basis dieser Gruppen- bzw. Gruppenmitgliederinformationen arbeitet das Übertragungsmodul. Dieses schätzt die Anzahl der potentiellen Empfänger einer Multicastnachricht mit Hilfe der Empfängeridentifikation und dem Wissen aus den lokalen Managementlisten. Eine Schätzung ist notwendig, denn die Information über die genaue Anzahl der aktuellen Gruppenmitglieder ist in einem DTN nicht verfügbar. Die Empfängeridentifikation ermittelt anhand eines Flag aus der Multicastnachricht sowie aus den Beitritts- und Austrittsinformationen von Gruppenmitgliedern diejenigen Knoten, die diese Multicastnachricht erhalten sollen. Basierend auf den ermittelten Empfängern einer Multicastnachricht und den erhaltenen Quittungen entscheidet der RMDA-Algorithmus, ob diese Multicastnachricht mit den zugehörigen Quittungen aus dem lokalen Nachrichtenspeicher gelöscht werden können. Zusätzlich ermöglicht die levelbasierte Speicherverwaltungsstrategie den Anwendern zwischen verschieden Leveln zu wählen und somit das Auslieferverhältnis von Multicastnachrichten zu Unicastnachrichten zu beeinflussen.

Der Vergleich mit dem BBR-Protokoll [Zhao u. a., 2005] und dem ECAM-Protokoll [Jin u. a., 2010] beweist, dass das RMDA-Protokoll im Hinblick auf den Auslieferungsgrad, diese Protokolle in verschieden Szenarien übertrifft.

Das RMDA-Protokoll bietet eine solide Grundlage für die Gruppenkommunikation. Trotzdem sind einige Ideen zur Optimierung des RMDA-Protokolls entstanden, konnten aber aus Zeitgründen nicht mehr umgesetzt werden.

Als nächster Schritt sollte das Protokoll in einem realen System implementiert und getestet werden.

Ergänzend zur manuellen Auswahl der Levels, sollte in der nächsten Entwicklungsphase eine automatische Levelauswahl erfolgen. Diese könnte sich an unterschiedlichen Nachrichtenaufkommen von Multicastnachrichten und Unicastnachrichten orientieren. Beispielsweise könnte die Standardeinstellung Level 1 sein und, wenn das Unicastnachrichtenaufkommen gegenüber den Multicastnachrichtenaufkommen auf 90 Prozent steigt, die Umschaltung auf Level 2 erfolgen. Dies weist den Unicastnachrichten einen höheren Speicherplatzanteil zu und erhöht die Chance der Zustellung zu den jeweiligen Zielknoten.

Eine positive Auswirkung auf den Auslieferungsgrad und die Verzögerung würde durch ein Priorisieren von Multicastnachrichten erreicht werden. Da die Reihenfolge des Nachrichtenaustauschs bisher nach dem FIFO-Prinzip erfolgt, wäre es sinnvoll auch für das RMDA-Protokoll eine Sortierung nach der Nachrichtenpriorität vorzunehmen und somit hoch priorisierte Nachrichten zuerst auszutauschen.

Sichere Datenübertragung spielt eine wichtige Rolle gerade bei BOS und deren Einsatz in Katastrophengebieten. Deshalb bietet das Thema Sicherheit Stoff für eine eigene weiterführende Arbeit.

Ähnlich aufwendig dürfte auch ein automatischer Gruppenbeitritt anhand von Attributen, beispielsweise Rollen, Dienste und Standorte, sein. Da stellen sich Fragen wie: Welche Attribute sind wichtig? Wie erfolgt die Adressierung?

Durch zusätzliche Berücksichtigung von Kontextinformationen, wie etwa die Abspeicherung geographischer Koordinaten während des Austauschs von Gruppenverwaltungsnachrichten, könnten Weiterleitungsentscheidungen auch Berücksichtigung bei der Speicherverwaltung finden. Beispielsweise könnten im RMDA-Algorithmus geogra-

phische Informationen so genutzt werden, dass wenn bei einer geringen Wahrscheinlichkeit, den oder die Zielknoten zu treffen, für den eine oder sogar mehrere Nachrichten gespeichert sind, diese Nachricht bzw. Nachrichten bevorzugt gelöscht werden.

Obwohl durch diese Arbeit ein anwendbares Protokoll entstanden ist, bietet dieses viel Spielraum für Erweiterungen. Es zeigt auch, das in dieser Richtung noch viel getan werden muss.

Dieses neuartige RMDA-Protokoll ermöglicht eine effiziente Gruppenkommunikation in einem Katastrophenszenario. Das ist die Basis für eine erfolgreiche Rettungsmission und hilft somit zukünftig Leben zu retten.

Abkürzungsverzeichniss

ACK	Acknowledgment
AHN	Ad-hoc Netz
ALM	Application Layer Multicast
ARBR	Adaptive Reinforcement-Based Routing
ARP	Address Resolution Protocol
BBR	Broadcast-Based Routing
BOS	Behörden und Organisationen mit Sicherheitsaufgaben
CAMR	Context Aware Multicast Routing
CERM	Controlled Epidemic Routing for Multicast
CGrAnt	Cultural Greedy Ant
CMD	Current-Member Delivery
DMO	Direct Mode Operation
DTCAST	Delay and Disruption Tolerant Multicasting Protocol
DTBR	Dynamic-Tree-Based Routing
DTN	Delay Tolerant Networking
DTNRG	Delay-Tolerant Networking Research Group
EBMR	Encounter-Based Multicast Routing
ECAM	Epidemic-based Controlled Flooding and Adaptive Multicast for Delay Tolerant Networks
EID	End-Point-Identifikation
FIMF	Ferry Initiated Message Ferrying
FIFO	First in First out
GBR	Group-Based Routing
GIS	Geographic Information System
GPL	General Public License
GPS	Global Positioning System
ID	Identifikator/Identifikation
IEEE	Institute of Electrical and Electronics Engineers

IP	Internetprotokoll
IPN	Interplanetares Internet
IPv4	Internet Protokoll Version 4
IPv6	Internet Protokoll Version 6
ISO	International Organization for Standardization
LAN	Local Area Network
MANET	Mobile Ad-hoc Network
MAC	Media-Access-Control
MCEpidemic	Multicast Epidemic
MIDTONE	Multicast In Delay Tolerant Networks
NASA	National Aeronautics and Space Administration
NIMF	Node Initiated Message Ferrying
ONE	Opportunistic Networking Environment
OS-Multicast	On-demand Situation-aware Multicasting
OSI	Open Systems Interconnection
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
QBMR	Quota Based Multicast Routing
QoS	Quality of Service
RFC	Requests for Comments
RMDA	Reliable Multicast over Delay Tolerant Mobile Ad Hoc Networks
SAR	Source Assured Reliability
SM	Social-Aware Multicast
SON	Self Organized Network
SPIDER	Security System for Public Institutions in Disastrous Emergeny Scenarios
SRW	Short Range Wireless
STBR	Static-Tree-Based Routing
TETRA	Terrestrial Trunked Radio
TTK	Time-To-Kill
TTL	Time-To-Live
UBR	Unicast-Based Routing
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VDTN	Vehicular Delay Tolerant Network

WAN	Wide Area Network
WKT	Well Known Text
WLAN	Wireless Local Area Network
YOID	Your Own Internet Distribution

Literaturverzeichnis

[Abdulla und Simon 2008] Abdulla, M. ; Simon, R.: Controlled Epidemic Routing for Multicasting in Delay Tolerant Networks. In: *16th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. Baltimore, MD, USA, September 2008, S. 1–10. – ISSN 1526-7539

[Afanasyev u. a. 2009] Afanasyev, Alexander ; Mayoral, Keith ; Zhu, Zhenkai ; Oh, Sonn Y.: DTCAST: Delay and Disruption Tolerant Multicasting Protocol. In: *Proceedings of the 11th Youth Technological Conference: High Technologies and Intellectual Systems, Moscow* (2009), April

[Albanna u. a. 2001] Albanna, Z. ; Almeroth, K. ; Meyer, D. ; Schipper, M.: *IANA Guidelines for IPv4 Multicast Address Assignments*. RFC 3171 (Best Current Practice). August 2001. – URL `http://www.ietf.org/rfc/rfc3171.txt`

[Antunes und Morla 2009] Antunes, Constantino ; Morla, Ricardo: Performance Analysis of a Hybrid Flooding-Probabilistic DTN Protocol using Logged Contact Data. In: *7th Conference on Telecommunications*. Santa Maria da Feira, Mai 2009

[Aschenbruck u. a. 2010] Aschenbruck, Nils ; Ernst, Raphael ; Gerhards-Padilla, Elmar ; Schwamborn, Matthias: BonnMotion: A Mobility Scenario Generation and Analysis Tool. In: *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium : ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010 (SIMUTools '10), S. 51:1–51:10. – URL `http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8684`. – ISBN 978-963-9799-87-5

[Atwood 2004] Atwood, J. W.: A Classification of Reliable Multicast Protocols. In: *Netwrk. Mag. of Global Internetwkg.* 18 (2004), Mai, Nr. 3, S. 24–34. – URL `http://dx.doi.org/10.1109/MNET.2004.1301019`. – ISSN 0890-8044

[Bärwald 2009] Bärwald, Werner: *expert Praxislexikon Kommunikationstechnologien: Netze - Dienste - Anwendungen.* expert, 2009. – 362 S. – ISBN 978-3816928430

[Berners-Lee u. a. 2005] Berners-Lee, T. ; Fielding, R. ; Masinter, L.: *Uniform Resource Identifier (URI): Generic Syntax.* RFC 3986 (INTERNET STANDARD). Januar 2005 (Request for Comments). – URL `http://www.ietf.org/rfc/rfc3986.txt`. – Updated by RFC 6874

[Berwanger u. a. 2015] Berwanger, Jörg ; Dennerlein, Birgitta ; Böcking, Hans-Joachim ; Oser, Peter ; Pfitzer, Norbert: *Gabler Wirtschaftslexikon, Stichwort: Fifo.* Springer Gabler Verlag. 2015. – URL `http://wirtschaftslexikon.gabler.de/Archiv/4105/fifo-v10.html`

[Board 2005] Board, IEEE-SA S.: *IEEE 802.15: WIRELESS PERSONAL AREA NETWORKS (PANs).* New York: Software Engineering Standard Committe of the IEEE Computer Society (Veranst.), 2005. – URL `http://standards.ieee.org/about/get/802/802.11.html`

[Board 2012] Board, IEEE-SA S.: *IEEE 802.11: Wireless LANs.* New York: Software Engineering Standard Committe of the IEEE Computer Society (Veranst.), 2012. – URL `http://standards.ieee.org/about/get/802/802.11.html`

[Borghoff und Schlichter 1995] Borghoff, Uwe ; Schlichter, Johann: *Rechnergestützte Gruppenarbeit - Eine Einfürung in Verteilte Anwendungen.* 1. Heidelberg : Springer-Verlag, 1995. – 422 S. – ISBN 978-3-642-97581-3

[Cerf u. a. 2007] Cerf, V. ; Burleigh, S. ; Hooke, A. ; Torgerson, L. ; Durst, R. ; Scott, K. ; Fall, K. ; Weiss, H.: *Delay-Tolerant Networking Architecture.* RFC 4838 (Informational). April 2007. – URL `http://www.ietf.org/rfc/rfc4838.txt`

[Choi und Shen 2010] Choi, Bong J. ; Shen, Xuemin: Distributed Clock Synchronization in Delay Tolerant Networks. In: *Communications (ICC), 2010 IEEE International Conference on Communications*, Mai 2010, S. 1–6. – ISSN 1550-3607

[Christiansen 2014] Christiansen, Jens: *TETRA (Terrestrial Trunked Radio).* Portal zur digitalen Funktechnik der Behörden und Organisationen mit Sicherheitsaufgaben. 2014. – URL `http://www.digitaler-bos-funk.de/index2.htm`

[Coulson u. a. 2005] Coulson, G. ; Grace, P. ; Blair, Gordon S. ; Duce, D. ; Cooper, C. ; Sagar, M.: A Middleware Approach for Pervasive Grid Environments. In: *UK-UbiNet/UK e-Science Programme Workshop on Ubiquitous Computing and e-Research*. Edinburg, Scotland, Mai 2005

[Diot 1995] Diot, Christophe: Reliability in Multicast Services and Protocols ; A Survey. In: Hasegawa, Toshiharu (Hrsg.) ; Pujolle, Guy (Hrsg.) ; Takagi, Hideaki (Hrsg.) ; Takahashi, Yutaka (Hrsg.): *Local and Metropolitan Communication Systems*. Springer US, 1995 (IFIP — The International Federation for Information Processing), S. 285–303. – URL `http://dx.doi.org/10.1007/978-0-387-34884-1_16`. – ISBN 978-1-4757-5672-2

[Eastlake und Abley 2013] Eastlake, D. ; Abley, J.: *IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters*. RFC 7042. Oktober 2013. – URL `http://www.ietf.org/rfc/rfc7042.txt`

[Eicker 2008] Eicker, Thomas: *Repeaer*. Ein kleines! Lexikon des Internet. August 2008. – URL `http://www.kleines-lexikon.de/w/r/repeater.shtml`. – [Online; Stand 21. Dezember 2015]

[Eicker 2011] Eicker, Thomas: *Gateway*. Ein kleines! Lexikon des Internet. Juli 2011. – URL `http://www.kleines-lexikon.de/w/g/gateway.shtml`. – [Online; Stand 20. Dezember 2015]

[Einstein 2011] Einstein, Albert: *Investigations on the Theory of the Brownian Movement*. BN Publishing, 2011. – ISBN 978-1607962854

[Ekman u. a. 2008] Ekman, Frans ; Keränen, Ari ; Karvo, Jouni ; Ott, Jörg: Working Day Movement Model. In: *1st ACM SIGMOBILE International Workshop on Mobility Models for Networking Research (MobilityModels) colocated with ACM SIGMOBILE MobiHoc'08*, May 2008, S. 33–40

[Elwhishi u. a. 2010] Elwhishi, A. ; Ho, Pin-Han ; Naik, K. ; Shihada, B.: ARBR: Adaptive reinforcement-based routing for DTN. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, Oktober 2010, S. 376–385

[Fall 2003] Fall, Kevin: A Delay-tolerant Network Architecture for Challenged Internets. In: *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, NY, USA :

ACM, 2003 (SIGCOMM '03), S. 27–34. – URL http://doi.acm.org/10.1145/863955.863960. – ISBN 1-58113-735-4

[Francis 1999] Francis, Paul: Yoid: Extending the internet multicast architecture. URL http://www.icir.org/yoid/, 1999. – Forschungsbericht

[Gao u. a. 2012] Gao, Wei ; Li, Qinghua ; Zhao, Bo ; Cao, Guohong: Social-Aware Multicast in Disruption-Tolerant Networks. In: *Networking, IEEE/ACM Transactions on* 20 (2012), Oktober, Nr. 5, S. 1553–1566. – ISSN 1063-6692

[Handley u. a. 2000] Handley, M. ; Floyd, S. ; Whetten, B. ; Kermode, R. ; Vicisano, L. ; Luby, M.: *The Reliable Multicast Design Space for Bulk Data Transfer*. RFC 2887 (Informational). August 2000. – URL http://www.ietf.org/rfc/rfc2887.txt

[Hinden und Deering 1998] Hinden, R. ; Deering, S.: *IP Version 6 Addressing Architecture*. RFC 2373 (Proposed Standard). Juli 1998. – URL http://www.ietf.org/rfc/rfc2373.txt

[Höher 2013] Höher, PeterAdam: Netzwerkcodierung. In: *Grundlagen der digitalen Informationsübertragung*. Springer Fachmedien Wiesbaden, 2013, S. 135–142. – URL http://dx.doi.org/10.1007/978-3-8348-2214-7_6. – ISBN 978-3-8348-1784-6

[Huang und Du 2015] Huang, Qiubo ; Du, Zhenzhen: *Computer Science and Applications 2015*. Kap. A method based on TDMA for improving the accuracy of neighbor list in VANET, S. 89–94, CRC Press, 2015. – URL http://dx.doi.org/10.1201/b18508-17

[Hußmann u. a. 2000] Hussmann, T. ; Kronenberg, H. ; Cimolino, U. ; Schneider, S.: *Einsatzstellen-Kommunikation: Planung, Organisation und Durchführung mit Praxisbeispielen für Stadt und Land*. Hüthig Jehle Rehm, 2000. – URL https://books.google.de/books?id=O3SKAAAACAAJ. – ISBN 9783609684307

[Irmscher 2011] Irmscher, Klaus: *Scriptum zur Lehrveranstaltung Rechnernetze*. 2011. – URL http://www.informatik.uni-leipzig.de/~irmscher/lehre/skripte/RechnernetzeScriptum_T1.pdf

[Jin u. a. 2010] Jin, Zhigang ; Wang, Jia ; Zhang, Sainan ; Shu, Yantai: Epidemic-Based Controlled Flooding and Adaptive Multicast for Delay Tolerant Networks. In: *Ubiquitous Intelligence Computing and 7th International Conference on Autonomic*

Trusted Computing (UIC/ATC), 2010 7th International Conference on, Oktober 2010, S. 191–194

[Keränen u. a. 2009] Keränen, A. ; Ott, J. ; Kärkkäinen, T.: The ONE Simulator for DTN Protocol Evaluation. In: *2nd International Conference on Simulation Tools and Techniques (SIMUTools)*. Rome, Italy : ICST, März 2009. – Article No. 55

[Lackes und Siepermann 2015a] Lackes, Richard ; Siepermann, Markus: *Gabler Wirtschaftslexikon, Stichwort: LAN*. Springer Gabler Verlag. 2015. – URL `http://wirtschaftslexikon.gabler.de/Archiv/74669/lokales-netz-v9.html`

[Lackes und Siepermann 2015b] Lackes, Richard ; Siepermann, Markus: *Gabler Wirtschaftslexikon, Stichwort: WAN*. Springer Gabler Verlag. 2015. – URL `http://wirtschaftslexikon.gabler.de/Archiv/76687/wan-v8.html`

[Lee u. a. 2008] Lee, Uichin ; Oh, Soon Y. ; Lee, Kang-Won ; Gerla, M.: RelayCast: Scalable Multicast Routing in Delay Tolerant Coalition Networks. In: *16th IEEE International Conference on Network Protocols (ICNP)*. Orlando, FL, USA, Oktober 2008, S. 218–227. – ISSN 1092-1648

[Li u. a. 2013] Li, Jianbo ; Jiang, Shan ; You, Lei ; Dai, Chenqu: A Location Based Controlled Epidemic Multicast Routing for Delay Tolerant Networks. In: *IJACT: International Journal of Advancements in Computing Technology* 5 (2013), Nr. 7, S. 972 – 982

[Lindgren u. a. 2012] Lindgren, A. ; Doria, A. ; Davies, E. ; Grasic, S.: *Probabilistic Routing Protocol for Intermittently Connected Networks*. RFC 6693 (Experimental). August 2012. – URL `http://www.ietf.org/rfc/rfc6693.txt`

[Lipinski u. a. 2015a] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Ad-hoc-Netz*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. Juni 2015. – URL `http://www.itwissen.info/definition/lexikon/Ad-hoc-Netzwerk-ad-hoc-network.html`. – [Online; Stand 20. November 2015]

[Lipinski u. a. 2015b] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Bluetooth*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. November 2015. – URL `http://www.itwissen.info/definition/lexikon/Bluetooth-Bluetooth.html`. – [Online; Stand 11. September 2015]

[Lipinski u. a. 2015c] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Dienst*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. August 2015. – URL `http://www.itwissen.info/definition/lexikon/Dienst-service.html`. – [Online; Stand 19. November 2015]

[Lipinski u. a. 2015d] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Heterogenes Netzwerk*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. Juni 2015. – URL `http://www.itwissen.info/definition/lexikon/Heterogenes-Netzwerk-heterogeneous-network.html`. – [Online; Stand 20. Dezember 2015]

[Lipinski u. a. 2015e] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Overhead*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. Dezember 2015. – URL `http://www.itwissen.info/definition/lexikon/Overhead-OH-overhead.html`. – [Online; Stand 10. Oktober 2015]

[Lipinski u. a. 2015f] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Routing*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. August 2015. – URL `http://www.itwissen.info/definition/lexikon/Routing-routing.html`. – [Online; Stand 19. November 2015]

[Lipinski u. a. 2015g] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Routing-Protokoll*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. November 2015. – URL `http://www.itwissen.info/definition/lexikon/Routing-Protokoll-routing-protocol.html`. – [Online; Stand 19. November 2015]

[Lipinski u. a. 2015h] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *Schicht*. Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. November 2015. – URL `http://www.itwissen.info/definition/lexikon/Schicht-layer.html`. – [Online; Stand 20. November 2015]

[Lipinski u. a. 2015i] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *SON (self organized network).* Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. November 2015. – URL `http://www.itwissen.info/definition/lexikon/SON-self-organized-network.html`. – [Online; Stand 18. November 2015]

[Lipinski u. a. 2015j] Lipinski, Klaus ; Lackner, Hans ; Laue, Oliver P. ; Kafka, Gerhard ; Niemann, Alexander ; Raasch, Eberhard ; Radonic, Bernhard Schoon A.: *WLAN.* Wissens-Portal ITwissen.info - DATACOM Buchverlag GmbH. November 2015. – URL `http://www.itwissen.info/definition/lexikon/wireless-lan-wlan-funk-lan.html`. – [Online; Stand 1. Dezember 2015]

[Lo und Luo 2012] Lo, Shou-Chih ; Luo, Nai-Wun: Quota-based Multicast Routing in Delay-Tolerant Networks. In: *15th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, September 2012, S. 544–548. – ISSN 1347-6890

[Machajewski 2015] Machajewski, Szymon: *What is a Computer Network? - Types & Definition, Chapter 62/Lesson 14.* Study.com. 2015. – URL `http://study.com/academy/lesson/what-is-a-computer-network-types-definition-quiz.html`

[Mankin u. a. 1998] Mankin, A. ; Romanow, A. ; Bradner, S. ; Paxson, V.: *IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols.* RFC 2357 (Informational). Juni 1998. – URL `http://www.ietf.org/rfc/rfc2357.txt`

[Mehta und Shah 2014] Mehta, Namita ; Shah, Mehul: Performance of Efficient Routing Protocol in Delay Tolerant Network: A Comparative Survey. In: *International Journal of Future Generation Communication & Networking* 7 (2014), März, Nr. 1, S. 151–158

[Narmawala und Srivastava 2009] Narmawala, Z. ; Srivastava, S.: MIDTONE: Multicast in delay tolerant networks. In: *Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference on*, August 2009, S. 1–8

[Nelson und Kravets 2010] Nelson, Samueil D. ; Kravets, Robin: For Members Only: Local and Robust Group Management in DTNs. In: *The Fifth ACM Workshop on Challenged Networks (CHANTS).* Chicago, Illinois, USA, September 2010, S. 5

[Onwuka 2011] Onwuka, E.: MANET: A Reliable Network in Disaster Areas. In: *Journal of Research in National Development* 9 (2011), Dezember, Nr. 5, S. 105–113. – URL http://www.transcampus.org/JORINDV9Dec2011/Jorind%20Vol9%20No2%20Dec%20Chapter17.pdf

[OpenJump 2011] OpenJump: *OpenJump*. 2011. – URL http://www.openjump.org/

[OpenStreetMap-Mitwirkende 2014] OpenStreetMap-Mitwirkende: *OpenStreetMap*. 2014. – URL http://www.openstreetmap.org

[Özcan u. a. 2011] Özcan, Abdulkadir ; Zizka, Jan ; Nagamalai, Dhinaharan: Recent Trends in Wireless and Mobile Networks. In: *Third International Conferences, WiMo 2011 and CoNeCo 2011* Bd. 162. Ankara, Turkey : Springer Berlin Heidelberg, Juni 2011

[Plate 2015] Plate, Jürgen: *Grundlagen Computernetze*. Hochschule München (FK 04). Juni 2015. – URL http://www.netzmafia.de/skripten/netze/netz7.html

[S. Symington 2006] S. Symington, K. S.: *Non-Custodial (Best-Effort) Multicasting Support in DTN*. August 2006. – URL http://www.dtnrg.org/docs/docs/specs/draft-symington-bundle-multicast-noncustodial-00.txt

[Schiller und Boigt 2003] Schiller, Jochen ; Boigt, Thiemo: *Next Generation Internet*. Freie Universität Berlin. 2003. – URL http://www.inf.fu-berlin.de/lehre/SS03/19531-V/lecture2.pdf

[Scott und Burleigh 2007] Scott, K. ; Burleigh, S.: *Bundle Protocol Specification*. RFC 5050 (Experimental). November 2007. – URL http://www.ietf.org/rfc/rfc5050.txt

[Shah u. a. 2003] Shah, R.C. ; Roy, S. ; Jain, S. ; Brunette, W.: Data MULEs: modeling a three-tier architecture for sparse sensor networks. In: *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, Mai 2003, S. 30–41

[Soares u. a. 2009] Soares, V. N. G. J. ; Farahmand, F. ; Rodrigues, J. J. P. C.: A layered architecture for vehicular delay-tolerant networks. In: *IEEE Symposium on Computers and Communications* IEEE (Veranst.), 2009, S. 122–127

[Spyropoulos u. a. 2005] Spyropoulos, Thrasyvoulos ; Psounis, Konstantinos ; Raghavendra, Cauligi S.: Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*. New York, NY, USA : ACM, 2005 (WDTN '05), S. 252–259. – URL http://doi.acm.org/10.1145/1080139.1080143. – ISBN 1-59593-026-4

[Srinivasan und Ramanathan 2010] Srinivasan, K. ; Ramanathan, P.: Reliable Multicasting in Disruption Tolerant Networks. In: *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dezember 2010, S. 1–5. – ISSN 1930-529X

[Vahdat und Becker 2000] Vahdat, Amin ; Becker, David: Epidemic Routing for Partially-Connected Ad Hoc Networks / Duke University (CS-200006). April 2000. – Forschungsbericht

[Vendramin u. a. 2012] Vendramin, Ana Cristina Barreiras K. ; Munaretto, Anelise ; Delgado, Myriam Regattieri de Biase da S. ; Viana, Aline C.: CGrAnt: A Swarm Intelligence-based Routing Protocol for Delay Tolerant Networks. In: *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA : ACM, 2012 (GECCO '12), S. 33–40. – URL http://doi.acm.org/10.1145/2330163.2330169. – ISBN 978-1-4503-1177-9

[Wietfeld 2012] Wietfeld, Christian: *Optimierte Kommunikation der Einsatzkräfte in Großschadenslagen*. BMBF-Innovationsforum: Zivile Sicherheit. Juni 2012. – URL http://www.bmbf.de/pubRD/B3-II_Wietfeld_Christian_Praesentation_2012.pdf

[Wikipedia 2014] Wikipedia: *Best Effort — Wikipedia, Die freie Enzyklopädie*. 2014. – URL https://de.wikipedia.org/w/index.php?title=Best_Effort&oldid=134825224. – [Online; Stand 13. November 2015]

[Wikipedia 2015a] Wikipedia: *Mobile ad hoc network — Wikipedia, Die freie Enzyklopädie*. 2015. – URL https://en.wikipedia.org/w/index.php?title=Mobile_ad_hoc_network&oldid=693938137. – [Online; Stand 6. Dezember 2015]

[Wikipedia 2015b] Wikipedia: *Multicast — Wikipedia, Die freie Enzyklopädie*. 2015. – URL https://de.wikipedia.org/w/index.php?title=Multicast&oldid=139130863. – [Online; Stand 11. Dezember 2015]

[Wittmann und Zitterbart 2000] Wittmann, Ralph ; Zitterbart, Martina: *Multicast Communication : Protocols, Programming, and Applications*. 1. Morgan Kaufmann Publishers, 2000. – ISBN 1558606459

[Wolff 2013] Wolff, Andreas H.: *Entwurf und Leistungsbewertung von Ad-hoc-Kommunikationsnetzen für den Katastrophenschutz*, TU Dortmund, Dissertation, 2013

[Xi und Chuah 2009] Xi, Yong ; Chuah, Mooi C.: An Encounter-based Multicast Scheme for Disruption Tolerant Networks. In: *Comput. Commun.* 32 (2009), Oktober, Nr. 16, S. 1742–1756. – URL `http://dx.doi.org/10.1016/j.comcom.2008.09.031`. – ISSN 0140-3664

[Yang u. a. 2005] Yang, Jeonghwa ; Chen, Yang ; Ammar, M. ; Lee, Chungkee: Ferry replacement protocols in sparse MANET message ferrying systems. In: *Wireless Communications and Networking Conference, 2005 IEEE* Bd. 4, März 2005, S. 2038–2044 Vol. 4. – ISSN 1525-3511

[Yang und Chuah 2006] Yang, Peng ; Chuah, Mooi C.: Context-aware Multicast Routing Scheme for Disruption Tolerant Networks. In: *3rd ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN) in conjunction with MSWiM 2006*. Torremolinos, Malaga, Spain, Oktober 2006, S. 66–73. – ISBN 1-59593-487-1

[Ye u. a. 2006] Ye, Qing ; Cheng, Liang ; Chuah, Mooi-Choo ; Davison, B.D.: OS-multicast: On-demand Situation-aware Multicasting in Disruption Tolerant Networks. In: *IEEE 63rd Vehicular Technology Conference (VTC 2006-Spring)*. Melbourne, Australia, Mai 2006, S. 96–100. – ISSN 1550-2252

[Ye u. a. 2009] Ye, Qing ; Cheng, Liang ; Chuah, Mooi C. ; Davison, Brian D.: Performance Comparison of Different Multicast Routing Strategies in Disruption Tolerant Networks. In: *Comput. Commun.* 32 (2009), Oktober, Nr. 16, S. 1731–1741. – URL `http://dx.doi.org/10.1016/j.comcom.2009.02.007`. – ISSN 0140-3664

[Yoon u. a. 2003] Yoon, J. ; Liu, M. ; Noble, B.: Random waypoint considered harmful. In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies* Bd. 2, March 2003, S. 1312–1321 vol.2. – ISSN 0743-166X

[Zhao u. a. 2004] Zhao, W. ; Ammar, M. ; Zegura, E.: A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In: *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA : ACM, 2004 (MobiHoc '04), S. 187–198. – URL `http://doi.acm.org/10.1145/989459.989483`. – ISBN 1-58113-849-0

[Zhao u. a. 2005] Zhao, Wenrui ; Ammer, Mostafa ; Zegura, Ellen: Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms. In: *WDTN '05 Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (2005), S. 268–275

Eigene Veröffentlichungen

[Begerow u. a. 2014a] Begerow, P. ; Krug, S. ; Schellenberg, S. ; Seitz, J.: Buffer Management for Reliable Multicast over Delay Tolerant Networks. In: *Mobile Ad-hoc and Sensor Networks (MSN), 2014 10th International Conference on*. Maui, Hawaii, USA, Dezember 2014, S. 171–178

[Begerow u. a. 2015] Begerow, P. ; Krug, S. ; Schellenberg, S. ; Seitz, J.: Robust reliability-aware buffer management for DTN multicast in disaster scenarios. In: *Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop on*, Oktober 2015, S. 274–280

[Begerow 2012] Begerow, Peggy: Multicast Management in verzögerungstoleranten Netzen in Verbindung mit Ad-hoc-Netzen. In: *Tagungsband / Ilmenauer TK-Manager-Workshop; 12 (Ilmenau)*, Univ.-Verl. Ilmenau, September 2012, S. 55–61. – ISBN 978-3-86360-036-5

[Begerow 2014] Begerow, Peggy: Zuverlässige Gruppenkommunikation in mobilen Ad-hoc-Netzen auf Basis eines verzögerungstoleranten Kommunikationsdienstes. In: *Tagungsband / Ilmenauer TK-Manager-Workshop; 13 (Ilmenau)*, Univ.-Verl. Ilmenau, September 2014, S. 88–93

[Begerow u. a. 2014b] Begerow, Peggy ; Krug, Silvia ; Renhak, Karsten ; Al-Rubaye, Atheer ; Seitz, Jochen: Delay Tolerant Handover for Heterogeneous Networks. In: *39th IEEE Conference on Local Computer Networks (LCN)*. Edmonton, Canada, September 2014, S. 370–373. – ISBN 978-1-4799-3780-6

[Begerow u. a. 2013] Begerow, Peggy ; Schellenberg, Sebastian ; Seitz, Jochen ; Finke, Thomas ; Schroeder, Juergen: Reliable Multicast in Heterogeneous Mobile Ad Hoc Networks. In: *Workshop on Self-Organized Communication in Disaster Scenarios (SoCoDiS) in conjunction with Networked Systems 2013*, März 2013

[Krug u. a. 2014a] Krug, S. ; Begerow, P. ; Al Rubaye, A. ; Schellenberg, S. ; Seitz, J.: A Realistic Underlay Concept for Delay Tolerant Networks in Disaster Scenarios. In: *Mobile Ad-hoc and Sensor Networks (MSN), 2014 10th International Conference on*, Dezember 2014, S. 163–170

[Krug u. a. 2014b] Krug, S. ; Siracusa, M.F. ; Schellenberg, S. ; Begerow, P. ; Seitz, J. ; Finke, T. ; Schroeder, J.: Movement patterns for mobile networks in disaster scenarios. In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*, Juni 2014, S. 1–6

[Schellenberg u. a. 2013] Schellenberg, S. ; Begerow, P. ; Hager, M. ; Seitz, J. ; Finke, T. ; Schroeder, J.: Implementation and Validation of an Address Resolution Mechanism using Adaptive Routing. In: *Information Networking (ICOIN), 2013 International Conference on*, Januar 2013, S. 95–100. – ISSN 1976-7684

[Schellenberg u. a. 2015] Schellenberg, Sebastian ; Krug, Silvia ; Finke, Thomas R. ; Begerow, Peggy ; Seitz, Jochen: Inter-Domain Routing and Name Resolution Using Border Nodes. In: *International Conference on Computing, Networking and Communications (ICNC 2015): Wireless Ad Hoc and Sensor Networks Symposium (WAHS)*. Anaheim, California, USA, Februar 2015, S. 950–956

Abbildungsverzeichnis

Tabellenverzeichnis